

RÉPUBLIQUE DÉMOCRATIQUE DU CONGO
ENSEIGNEMENT SUPÉRIEUR ET UNIVERSITAIRE
Institut Supérieur Pédagogique de Bukavu
I.S.P/BUKAVU



BP : 854/BUKAVU

Section des Sciences Exactes

Département de Mathématique - Physique

Sur les méthodes numériques d'optimisation non linéaire

*Travail de fin de cycle présenté pour l'obtention
du diplôme de gradué en pédagogie appliquée*

Option : Mathématique-Physique

Présenté par

NYANDU KAGARABI Emmanuel

Dirigé par

Dr Justin BUHENDWA NYENYEZI

Année académique 2018 - 2019

Table des matières

Introduction	1
0.1 Contexte	1
0.2 Problématique	1
0.3 Hypothèses	2
0.4 Importance du sujet	2
0.5 Objectifs du travail	3
0.6 Choix et intérêt du sujet	3
0.7 Délimitation du sujet	3
0.8 Méthodologie de travail	3
0.9 Subdivision du travail	4
0.10 Difficultés rencontrées	4
1 Généralités sur l'optimisation numérique	5
1.1 Motivation	5
1.1.1 Qu'est-ce qu'un problème d'optimisation ?	6
1.1.2 Exemples de problèmes d'optimisation non linéaire	7
1.2 Définition des concepts fondamentaux	7
1.2.1 Ouverts et fermés de \mathbb{R}^n	7
1.2.2 Notion de minimum, maximum, infimum et supremum	8
1.2.3 convexité et concavité	10
1.2.4 Différentiabilité	12
1.2.5 Signe d'une matrice	14
1.3 Conditions d'Optimalité	16
1.3.1 Optimisation sans contrainte	17
1.3.2 Optimisation avec contraintes	20
1.4 Convergence et vitesse de convergence d'un algorithme	28
1.5 Existence et unicité d'un point minimum	29
2 Méthodes numériques d'optimisation sans contrainte	31
2.1 Méthode de Newton locale	31
2.2 Problèmes quadratiques	32
2.2.1 Résolution directe	32
2.2.2 Méthode des gradients conjugués	33
2.3 Méthodes de descente et recherche linéaire	35
2.3.1 Principe général des méthodes de descente	35
2.3.2 Recherche linéaire	35
2.3.3 Méthode de la plus forte pente	36
2.3.4 Méthode de Newton avec recherche linéaire	37
2.4 Méthodes de région de confiance	38

2.4.1	Résolution du sous problème de région de confiance	40
2.4.2	Méthode dogleg	40
2.4.3	Méthode de Steihaug-Toint	41
2.4.4	Calcul du rayon de la région de confiance	42
2.5	Méthodes quasi-Newton	43
2.5.1	Méthode BFGS	44
2.5.2	Mise à jour symétrique de rang 1	45
3	Méthodes numériques d'optimisation avec contraintes	47
3.1	Méthode de Newton avec contraintes	47
3.1.1	Méthode du gradient projeté	48
3.1.2	Gradient projeté préconditionnée	50
3.2	Lagrangien augmenté	51
3.2.1	Pénalisation Lagrangienne	51
3.2.2	Pénalisation quadratique	52
3.2.3	Double pénalisation	52
3.3	Programmation quadratique séquentielle	55
3.3.1	Programmation quadratique séquentielle locale	56
3.3.2	Algorithme globalement convergent	58
4	Applications numériques de quelques méthodes d'optimisation sans contrainte	61
4.1	Méthode de la plus forte pente	62
4.1.1	Tableau d'itérations	62
4.1.2	Graphique de la fonction	63
4.1.3	Code octave I	63
4.2	Méthode de Newton	64
4.2.1	Tableau d'itérations	64
4.2.2	Graphique de la fonction	64
4.2.3	Code octave II	65
4.3	Comparaison entre la méthode de Newton et celle de la plus forte pente . .	65
4.3.1	Graphiques des deux méthodes	65
4.3.2	Code Octave III	66
	Conclusion	67
	Bibliographie	68

Épigraphe

“L’honneur est donc pour vous , qui croyez. Mais, pour les incrédules, la pierre qu’ont rejetée ceux qui bâtaient est devenue la principale de l’angle, et une pierre d’achoppement et un rocher de scandale”. (1Pierre 2 : 7-8)

“Ne refuse pas un bienfait à celui qui y a droit, quand tu peux le faire, ne dis pas à ton prochain : Vas et reviens demain je te donnerai ! quand tu as de quoi donner”. (Proverbes 3 : 27-28)

“Dieu, qui connaît le mieux les capacitésR des hommes, cache ses mystères aux sages et aux prudents de ce monde, et les révèle aux petits enfants”. (Isaac Newton)

“La mathématique est l’art de donner le même nom à des choses différentes”. (Henry Poincaré)

“Quiconque arrête d’apprendre est vieux, qu’il ait 20 ou 80 ans. Quiconque continue à apprendre demeure jeune. La plus grande chose dans la vie, c’est garder son esprit jeune”. (Henry Ford)

“Soit A un succès dans la vie. Alors $A = x + y + z$, où x = travailler, y = s’amuser et z = se taire”. (Albert Einstein)

“La philosophie est écrite dans ce grand livre, l’univers, qui ne cesse pas d’être ouvert devant nos yeux. Mais ce livre ne peut se lire si on ne comprend pas le langage et on ne connaît pas les caractères avec lesquels il est écrit. Or, la langue est celle des mathématiques, et les caractères sont triangles, cercles et d’autres figures géométriques. Si on ne les connaît pas, c’est humainement impossible d’en comprendre même pas un seul mot. Sans eux, on ne peut qu’aller à la dérive dans un labyrinthe obscur et inextricable”. (Galileo Galilei, dit Galilée, Pise 1564-1642 Arcetri Savant et écrivain italien, Il Saggiatore, Rome, 1623)

Dédicace

À ma famille, ami(e)s et connaissances !.

In memorium

À notre mère Berthe CIKURU que le très haut a précocement arraché à notre attachement avant de jouir du fruit de son travail, son affection et sa tendresse.

Quoi qu'endormi à l'attente du christ, que la terre des nos ancêtres lui demeure douce.

Remerciements

Que gloire soit rendue à l'éternel Dieu, tout puissant, créateur du ciel et de la terre, omnipotent, omniscient et miséricordieux lui qui nous a garanti courage, protection et santé tout au long de notre parcours.

Le présent travail est certes le fruit de plusieurs personnes qui, de près ou de loin y ont contribué; ce qui nous pousse à remercier tous ceux dont l'assistance intellectuelle, morale et surtout matérielle a permis sa réalisation.

De façon exceptionnelle, nous remercions vivement le Docteur Justin BUHENDWA NYENYEZI pour avoir accepté de diriger ce travail malgré son emploi de temps très surchargé. Qu'à travers lui tout le corps professoral de l'ISP BUKAVU en général et plus particulièrement tous les enseignants des départements de Mathématiques-Physique et Physique-Technologie trouvent ici l'expression de notre profonde gratitude. Nous rendons hommage aux CT Jean KYALULA et MUSHIWALYAHYAGE ZALUKA pour avoir contribué tant moralement que matériellement à notre parcours dans ce premier cycle. De même au CT MBILIZI Déogratias du département d'Informatique de gestion.

Nos remerciements pour leur soutien et encouragement, à nos parents KAGARABI Joseph et NZIGIRE BUROKO Espérance.

Un mot de remerciement très spécifique adressé d'abord à nos frères AKSANTI KAGARABI Edouard et BIRABALUGE KAGARABI Jean Paul pour leurs efforts et sacrifices consentis en ma faveur. En suite à nos Sœurs NABINTU KAGARABI Sylvie, Bernadette KAGARABI, NYOTA KAGARABI Angélique, FAIDA KAGARABI Pascasie, ISHARA KAGARABI Rosette, NZIGIRE KAGARABI Christine, NSIMIRE KAGARABI Berline, Hosanna KAGARABI, BINJA KAGARABI Ghyslaine, Nadine KAGARABI. Et enfin à nos familiers KOKO KAZIBI Justin, Ahadi Buroko Justin, Ushindi, Clotilde. Ainsi qu'à notre oncle BIRABALUGE BUROKO Isaac.

Notre gratitude s'adresse également à nos camarades représentés ici par CEKUMI LUGULIKA Eustache, CIKANYI BAHIMUZI Julien, BYAMUNGU BENGEBARHI Claude, BULONZA RUKARA Adelin, SAFARI MUTABAZI Michael, AKILIMALI Paul, AZENE Willy, MUGOLI BADESIRE Francine, ISHARA KABOBYA François, MATABARO Horomicidas, ASIFIWE MIHIGO Ruphin, BUNANI Christian, etc.

Nous remercions également nos amis et collaborateurs représentés par MUGISHO BALAGA Sammy, AGANZE MATABARO Mario, NYORHA NZIGIZI Soif, MBONEKUBE Rogger, HABANAWEMA CIZA Chistelle, BULONZA MATABARO Rachel, MULUME BAGALWA Pascal, BYAMUNGU SANVURA Raymond, MASIRIKA Adrien, Daniella, Patient Huguens, etc.

Enfin, que tous ceux dont les noms ne sont pas ci-haut cités, qu'ils ne se sentent pas oubliés mais qu'ils trouvent à travers cette rédaction l'expression de notre profonde affection et gratitude envers eux.

NYANDU KAGARABI Emmanuel

Introduction

0.1 Contexte

En mathématiques tout comme dans d'autres sciences, il est parfois question de déterminer l'extremum d'une fonction traduisant un problème quelconque. Or qui parle de la détermination de l'extremum fait généralement d'abord référence au choix d'une méthode de calcul à utiliser afin d'aboutir à cet extremum, ensuite savoir l'appliquer et enfin interpréter le résultat obtenu. Généralement les méthodes de résolution sont de deux catégories notamment celles dites "analytiques" et d'autres dites "numériques". Dans la pratique, même le calcul des dérivées premières d'une fonction s'avère parfois ardu et celui des dérivées secondes inextricable. Cependant, les méthodes analytiques peuvent fournir une réponse exacte au problème d'optimisation, mais dans la plupart des cas elles sont inutilisables, surtout lorsque le problème est de grande taille ou en dimension élevée ($\dim > 3$). Ces facteurs peuvent alors amener à choisir une méthode itérative, car elle est en général beaucoup plus rapide et contourne certains problèmes de calcul.

Les méthodes numériques d'optimisation sont aussi itératives et sont utilisées dans la recherche d'extrema d'une fonction continue et au moins une fois différentiable, elles peuvent être utilisées pour une résolution efficiente d'un problème de grande taille. Les algorithmes¹ d'optimisation sont des outils importants pour les ingénieurs, mais difficiles à utiliser. En effet, aucun d'entre eux n'est universel et une bonne compréhension des différentes méthodes est nécessaire pour identifier, voire adapter les outils aux besoins des applications ; D'où la nécessité d'une étude rigoureuse des méthodes numériques d'optimisation non linéaire. C'est pourquoi notre préoccupation dans ce travail sera alors d'élucider des classes de méthodes permettant de trouver solution aux problèmes de la forme

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

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$, appelée fonction objectif est une fonction continue et au moins une fois différentiable qui peut être soumise à des contraintes ou pas.

0.2 Problématique

Les méthodes numériques de résolution de divers problèmes d'extremum ont pris ces dernières années un bel essor. Cet intérêt n'est nullement fortuit, il reflète le rôle de premier plan que les problèmes d'extremum jouent dans les applications. Cependant les exigences formulées à l'égard de nouveaux algorithmes ne sont pas les mêmes qu'il y a dix ou quinze années où chaque nouvelle procédure de calcul pour tel ou tel problème de minimisation a été accueillie avec intérêt, il ne suffit actuellement pas de construire un algorithme, il faut montrer de plus qu'il l'emporte sur ceux connus. On a donc à comparer l'efficacité de différents algorithmes, problème qui ne se résout pas hélas si facilement. En effet, on ne compare qu'en se basant sur un critère, or il y en a plus d'un (par exemple le temps de calcul, l'occupation de mémoire nécessaire, ...) et il arrive qu'on demande d'évaluer

1. **Un algorithme** est une méthode de calcul précise, rigoureuse, qui pour chaque donnée, fournit un résultat en un nombre fini d'étapes élémentaires. ([1] : p6)

un algorithme selon plusieurs critères assez contradictoires. Les algorithmes à développer dans ce travail sont essentiellement partis du critère de précision du résultat et de celui de vitesse de convergence du processus itératif, cependant même en restant dans ce cadre étroit, on ne peut ordonner univoquement tous les algorithmes ni en indiquer le meilleur et le pire, le fait est qu'on obtient les estimations de la vitesse de convergence pour des classes de problèmes, non pas pour des problèmes isolés, et un algorithme mauvais pour une classe peut s'avérer efficace pour une autre, plus restreinte. Ce travail va essayer d'explicitier des méthodes d'optimisation non linéaire et comparer certaines d'entre elles pour un problème choisi.

0.3 Hypothèses

En optimisation numérique, il existe des problèmes pour lesquels il n'est pas facile de formuler des hypothèses. En effet, il est si pas impossible, alors il est difficile de prévoir à l'avance la solution la plus exhaustive.

En particulier, d'une part nous affirmons l'explicitation des méthodes d'optimisation non linéaire et d'autre part, signalons qu'il n'y a pas de méthode ou d'algorithme qui soit universel pour résoudre un problème d'optimisation non linéaire, c'est pourquoi l'utilisateur serait censé donc posséder tout un arsenal d'algorithmes pour être à mesure de faire face à chaque problème proposé. Mais nous estimons que des méthodes d'optimisation sans contrainte telles que la méthode de la plus forte pente, celle de Newton, etc. Et d'autres d'optimisation avec contraintes comme la méthode du gradient projeté, celle du lagrangien augmenté, etc. aideraient à la résolution d'un problème d'optimisation non linéaire respectivement sans ou avec contraintes.

0.4 Importance du sujet

Des nombreuses situations issues des mathématiques, des sciences expérimentales ou de la vie sociale conduisent à la recherche et à l'étude des extrema des fonctions. Cette recherche d'extrema occupe une place importante en mathématiques et dans les sciences en général car elle intervient dès lors qu'on cherche à optimiser une certaine quantité. Parmi de nombreux exemples on peut citer :

- En **Économie** : on cherchera à maximiser les profits et à minimiser les coûts.
- En **Optique** : les lois de la réflexion et de la réfraction découlent d'une minimisation du temps de parcours de la lumière.
- En **Mécanique** : l'étude de la stabilité d'un système est liée à des extrema. Par exemple ; un théorème classique concernant ce problème, le théorème de LEJEUNE-DIRICHLET, a pour hypothèse que la fonction de force possède un maximum local strict.
- etc.

Par suite, ce travail s'avère important en ce sens qu'il va proposer des méthodes numériques aux intéressés dans le but de leur faciliter la détermination fructueuse d'extremum.

0.5 Objectifs du travail

Le présent travail poursuit un triple objectif :

1. Proposer un état de l'art explicite des applications et de l'efficacité des méthodes numériques d'optimisation non linéaire.
2. Doter l'utilisateur d'un arsenal des méthodes numériques et algorithmes lui permettant de faire face à une classe importante des problèmes d'optimisation non linéaire.
3. Expérimenter certaines des méthodes en implémentant dans le langage octave un programme d'optimisation numérique.

0.6 Choix et intérêt du sujet

Notre choix a porté sur les méthodes numériques d'optimisation non linéaire car après avoir constaté la rareté, voire-même l'absence de tels travaux dans nos bibliothèques locales, nous pensons montrer avec ce travail qu'à part les anciennes méthodes dites analytiques permettant la détermination d'extrema d'une fonction donnée, qui prennent du temps et parfois présentant des limites dans le calcul, il existe alors d'autres dites *numériques* apparaissant modernes, rapides et surtout plus performantes.

Quoi que, généralement, elles exigent l'usage exclusif et systématique des supports informatiques comme le langage de programmation MATLAB/Octave, etc.

Ce sont donc de tels aspects qui ont motivé notre choix et intérêt pour ce sujet.

0.7 Délimitation du sujet

Une étude sur les méthodes numériques d'optimisation non linéaire serait très large à tel point que le temps nous imparti pour ce travail serait insuffisant pour en étudier tous les aspects. C'est pourquoi nous nous limiterons sur la présentation de quelques unes des méthodes numériques d'optimisation non linéaire (dites "Méthodes de descente "), qui sont évidemment itératives c'est-à-dire des algorithmes permettant de construire une suite finie ou infinie des points $x_k, k \in \mathbb{N}$ reliés par la relation $x_{k+1} = x_k + \alpha_k d_k$ où x_k est la valeur de x à l'étape k et α_k en est le pas dans la direction d_k . Nous traitons particulièrement les méthodes d'optimisation non linéaire différentiable en deux volets : d'une part les méthodes d'optimisation sans contrainte et d'autre part les méthodes avec contraintes.

Il existe d'autres types de problèmes d'optimisation qui se traitent différemment et qui n'ont pas fait l'objet de ce travail : problèmes d'optimisation linéaire (qui se traitent avec des méthodes différentes, comme la méthode du simplexe), problèmes d'optimisation discrète (lorsque les variables ne sont pas continues).

0.8 Méthodologie de travail

L'atteinte de nos objectifs s'est basée d'une part sur la démarche documentaire ; celle-ci nous a permis de sélectionner, rassembler, ranger et organiser les données relatives au sujet de recherche.

Et d'autre part, une méthode expérimentale par simulations numériques dans le langage de programmation octave nous a (à son tour) permis d'expérimenter certaines des méthodes développées.

0.9 Subdivision du travail

Hormis l'introduction et la conclusion générale, ce travail comprend quatre chapitres :

Le premier s'intitule "Généralités sur l'optimisation numérique" et consiste à donner des notions de base en optimisation numérique jugées nécessaires pour le développement des autres chapitres du travail.

Le deuxième chapitre est intitulé "Méthodes numériques d'optimisation sans contrainte". Dans ce chapitre, nous y détaillons en long et en large des méthodes numériques conçues spécifiquement pour minimiser des fonctions en l'absence des contraintes, et qui serviront d'appui au chapitre 3 étant donné qu'un problème avec au moins une contrainte se réduit généralement à des sous-problèmes sans contrainte paraissant faciles à résoudre. Il s'agira donc d'identifier une direction de descente et de calculer un pas approprié dans cette direction de telle sorte qu'il y ait diminution suffisante de la fonction objectif.

Le troisième chapitre est intitulé "Méthodes numériques d'optimisation avec contraintes" ; s'appuyant aux méthodes du précédent chapitre, dans ce troisième nous allons élucider des méthodes spécifiques à la résolution des problèmes d'optimisation pour lesquels la fonction d'utilité est soumise au moins à une contrainte². Deux objectifs doivent être atteints à ce niveau : diminuer la valeur de la fonction objectif et vérifier les contraintes. Néanmoins ces deux objectifs sont souvent en conflit dans le cas où, pour essayer de vérifier les contraintes, il faut en général augmenter la valeur de la fonction.

Enfin dans le quatrième et dernier chapitre intitulé "Applications numériques de quelques méthodes d'optimisation sans contrainte" nous allons passer à une expérimentation des certaines méthodes développées dans le deuxième chapitre.

0.10 Difficultés rencontrées

Pendant la réalisation de ce travail, nous nous sommes heurté à certaines difficultés dont le manque des notions de base en mathématiques numériques (appliquées) car sauf indication contraire jusqu'en ces jours le programme suivi dans notre département privilégie l'approche analytique et puriste des mathématiques.

Mais aussi le manque d'ouvrages dans nos bibliothèques locales surtout adaptés au domaine de recherche ont entravé l'évolution du travail à un moment donné, nous étions alors obligé de fréquenter l'internet régulièrement ; raison pour la quelle beaucoup d'ouvrages utilisés étaient en version électronique.

C'est ainsi qu'un grand effort d'auto-apprentissage a été fourni de notre part afin de réaliser ce travail.

2. **Une contrainte** du problème d'optimisation est une condition que doivent respecter les vecteurs de décision du problème. ([1] : p11)

Chapitre 1

Généralités sur l'optimisation numérique

(L'essentiel de ce chapitre a été inspiré des références bibliographiques suivantes : [14] : pp41-179, [7] : pp7-45, [4] : pp9-26, [16] : pp5-29, [5] : pp17-19 , [17] : pp5-19 et [8] : pp2-31)

Ce présent chapitre constitue la fondation de notre travail car il s'avère un outil incontournable pour la compréhension du reste du travail.

En effet ; dans la première section nous essayons d'abord d'y présenter un peu de motivation relative au domaine de recherche, qui est évidemment l'optimisation numérique non linéaire.

Dans la deuxième ; nous y détaillons les définitions de certains concepts fondamentaux estimés nécessaires et qui serviront d'explicitation aux autres chapitres.

Par ailleurs, dans la troisième ; les conditions d'optimalité permettant de dégager la caractérisation d'une solution au problème d'optimisation considéré, elles sont jugées capitales pour la compréhension et l'analyse de différentes méthodes et algorithmes à développer dans les deuxième et troisième chapitres.

Dans la quatrième section nous présentons la convergence et vitesse de convergence d'un algorithme, celles-ci vont nous donner une idée sur la façon dont certains algorithmes génèrent des résultats ainsi que l'interprétation de ces algorithmes, car nous en parlerons fréquemment dans les chapitres suivants.

Dans la cinquième et dernière section nous y dégageons les résultats d'existence et d'unicité d'un point minimum à un problème donné. En voici alors l'essentiel :

1.1 Motivation

L'optimisation et particulièrement l'optimisation numérique a connu un essor important ces dernières années avec l'avènement de l'ordinateur, elle est souvent l'ultime étape de l'analyse numérique où après avoir étudié un phénomène physique, l'avoir mis en équation, avoir étudié ces équations et avoir montré que l'on pouvait calculer les solutions avec un ordinateur, on commence à optimiser le système en changeant certains paramètres pour changer la solution dans un sens désiré.

En nous intéressant des notions de base de l'optimisation et pour se fixer les idées, imaginons-nous un étudiant qui doit réviser ses examens, nous supposons qu'il a trois matières à passer et qu'il révise x_i heures sur la i^e matière, pour $i = 1, 2, 3$.

Nous supposons alors que la note n_i de la matière i dépend du temps passé x_i . Pour nous fixer les idées supposons que $n_1(x_1) = 3x_1$, $n_2(x_2) = x_2^2$, $n_3(x_3) = x_3(100 - x_3)$. Cet étudiant veut passer au plus 10 heures au total à réviser et veut optimiser sa moyenne totale.

On note $x = (x_1, x_2, x_3)$ le triplet qui correspond aux heures passées à travailler, et aussi

$$f(x) = \frac{1}{3} (3x_1 + x_2^2 + x_3(100 - x_3)) \quad (1.1)$$

sa moyenne. D'après les données du problème, x ne peut pas prendre n'importe quelles valeurs réelles, nous disons que le problème est **contraint**.

En effet, x doit appartenir à X , l'ensemble des contraintes.

$$X = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \text{ tel que } \forall i, x_i \geq 0 \text{ et } x_1 + x_2 + x_3 \leq 10\}. \quad (1.2)$$

On note le problème de l'étudiant

$$\max_{x \in X} f(x). \quad (1.3)$$

En généralisant notre exemple à la modélisation mathématique du problème d'optimisation, on distingue trois étapes :

1. Identification des variables de décision (désignées par le vecteur $x \in \mathbb{R}^3$ dans notre exemple) : ce sont les paramètres sur lesquels l'utilisateur peut agir pour faire évoluer le système considéré.
2. Définition d'une fonction objectif ou fonction coût ou fonction économique ou encore fonction d'utilité (la fonction f dans notre exemple) permettant d'évaluer l'état du système (ex : rendement, performance, etc.)
3. Description des contraintes imposées aux variables de décision (l'ensemble X dans notre exemple). ([8] : pp5-6)

1.1.1 Qu'est-ce qu'un problème d'optimisation ?

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum (de l'optimum donc) d'une fonction donnée. On peut aussi trouver des problèmes d'optimisation pour lesquels les variables de la fonction à optimiser sont contraintes d'évoluer dans une certaine partie de l'espace de recherche ; dans ce cas, on a une forme particulière de ce qu'on appelle **problème d'optimisation sous contraintes**. ([1] : p9)

En d'autres termes, le problème d'optimisation consiste à déterminer les variables de décision conduisant aux meilleures conditions de fonctionnement du système. (ce qui revient à **maximiser** ou à **minimiser** la fonction coût) tout en respectant les contraintes d'utilisation ; pour notre exemple elles sont définies par l'équation (1.2).

Ainsi le problème se note

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1.4)$$

ou

$$\max_{x \in \mathbb{R}^n} f(x) \quad (1.5)$$

selon qu'on cherche à minimiser ou respectivement à maximiser le système considéré.

Et bien, en particulier dans notre présent travail de recherche, notre attention sera surtout plus spécifiquement focalisée sur les méthodes de **minimisation** d'une fonction non linéaire étant donné que la **maximisation** découle de la minimisation de $-f$.

D'où

$$\min_{x \in \mathbb{R}^n} f(x) = - \max_{x \in \mathbb{R}^n} (-f(x)) \quad (1.6)$$

ou encore

$$\max_{x \in \mathbb{R}^n} f(x) = - \min_{x \in \mathbb{R}^n} (-f(x)). \text{ ([9] : pp5-6)}$$

1.1.2 Exemples de problèmes d'optimisation non linéaire

1. Soit $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ qui, au couple (a, b) associe le réel $a^2 + b^2 + ab - 5a + 2b$.
C'est un problème sans contrainte, chaque variable a, b varie dans \mathbb{R} tout entier.

2. Problème de production

Une usine a besoin de n produits a_1, a_2, \dots, a_n en quantité x_1, \dots, x_n pour fabriquer un produit fini b en quantité $h(x_1, \dots, x_n)$. Soit p_i le prix unitaire du produit i et p le prix de vente du produit fini. On désire calculer la production optimale.

Il s'agit donc de maximiser la fonction

$$ph(x_1, \dots, x_n) - \sum_{i=1}^n x_i p_i.$$

3. Un problème avec contrainte de type égalité

On veut trouver le parallélépipède de volume maximal et de surface donnée. Soient a, b, c les différentes longueurs des côtés.

On cherche donc à maximiser la fonction abc sous la contrainte $2(ab + ac + bc) = S$.

4. Un problème avec contrainte de type inégalité

Une firme fabrique des objets A et des objets B en utilisant des matières premières m_1 et m_2 . Pour fabriquer une unité de A, il faut 2 unités de m_1 et 1 de m_2 . Et pour fabriquer une unité de B, il faut 1 unité de m_1 et 2 de m_2 . On dispose de 8 unités de m_1 et 7 de m_2 . Enfin, le bénéfice est de 4 par unité de A et de 5 par unité de B. Le patron de la firme voudrait optimiser son bénéfice.

Il s'agit donc de maximiser sur S la fonction $4a + 5b$ où a (resp. b) représente la quantité de produit A (resp. B) fabriqué avec S défini par

$$S = \{(a, b) \in \mathbb{N}^2 \text{ tel que } 2a + b \leq 8, a + 2b \leq 7\}. \text{ ([8] : pp3-4).}$$

Ceci étant, nous passons alors à la définition des notions nécessaires dans cette suivante section.

1.2 Définition des concepts fondamentaux

Dans cette section, nous élucidons certains concepts fondamentaux voir même des exemples d'illustration dans la mesure du possible. En voici alors l'essentiel :

1.2.1 Ouverts et fermés de \mathbb{R}^n

Soient $x \in \mathbb{R}^n$ et $r > 0$. On appelle boule ouverte de centre x et de rayon r l'ensemble :

$$B(x, r) = \{y \in \mathbb{R}^n \text{ tel que } \|y - x\| < r\} \quad (1.7)$$

Définition 1.1. Ouvert, fermé ([8] : p7)

Un ensemble O de \mathbb{R}^n est dit ouvert si pour tout $x \in O$, il existe une boule ouverte $B(x, r)$ de centre x et de rayon r incluse dans O . Un ensemble F de \mathbb{R}^n est dit fermé si son complémentaire est un ouvert.

L'interprétation géométrique de "Ouvert" et "Fermé" est qu'un ensemble est ouvert si et seulement si il ne contient aucun point de sa frontière, par ailleurs il sera dit fermé si et seulement si il contient tous les points de sa frontière, les ensembles qui contiennent une partie de leur frontière mais pas l'intégralité ne sont ni ouverts, ni fermés. Signalons cependant que cette définition fournit ci-haut est celle classique qui construit la topologie usuelle. Néanmoins on dirait en d'autres termes qu'un ensemble $F \subset \mathbb{R}^n$ est un fermé si et seulement si toute suite convergente d'éléments de F a sa limite dans F .

1.2.2 Notion de minimum, maximum, infimum et supremum

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

Définition 1.2 (Majorant/Minorant).

Soit Y un sous-ensemble de \mathbb{R} , $m \in \mathbb{R} \cup \{-\infty, +\infty\}$ est un minorant de Y si et seulement si m est inférieur ou égal à tous les éléments de Y . (m minorant $\Leftrightarrow \forall x \in Y, m \leq x$).

Tandis que $M \in \mathbb{R}$ est un majorant de Y si et seulement s'il est supérieur ou égal à tous les éléments de Y . Ainsi, (M majorant $\Leftrightarrow \forall x \in Y, M \geq x$).

Si Y admet un minorant (resp. un majorant) fini alors il est minoré (resp. majoré). Par ailleurs, Y est borné si et seulement s'il est minoré et majoré. Ainsi, (Y est borné $\Leftrightarrow \forall x \in Y, |x| \leq M$).

Définition 1.3 (Infimum/Supremum).

Soit $Y \subset \mathbb{R}$, l'infimum $\inf(Y) \in \mathbb{R} \cup \{-\infty, +\infty\}$ de Y est le plus grand des minorants alors que le supremum $\sup(Y) \in \mathbb{R} \cup \{-\infty, +\infty\}$ de Y est le plus petit des majorants.

Définition 1.4 (Fonction bornée inférieurement).

Soit une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et un ensemble $Y \subset \mathbb{R}$. La fonction f est bornée inférieurement sur Y s'il existe un réel M tel que $f(x) \geq M, \forall x \in Y$.

Par ailleurs, l'infimum de f sur Y est noté $\inf_{y \in Y} f(y)$. Il est tel que, $\forall x \in Y, \inf_{y \in Y} f(y) \leq f(x)$ et $\forall M > \inf_{y \in Y} f(y), \exists x \in Y$ tel que $f(x) < M$.

Définition 1.5 (Minimum/Maximum).

Soit $Y \subset \mathbb{R}$, l'infimum de Y est appelé minimum si et seulement si $\inf(y) \in Y$. Tandis que le supremum de Y est appelé maximum si et seulement si $\sup(y) \in Y$. Dans ce cas, on les note respectivement $\min(Y)$ et $\max(Y)$.

Exemple 1.1. Si on se donne $Y =]0, 1]$, alors l'ensemble des minorants de Y est $] -\infty, 0]$ et l'ensemble des majorants est $[1, +\infty]$. On en déduit que l'infimum vaut 0 et le supremum vaut 1. Comme $1 \in Y$ alors c'est aussi le maximum de Y , cependant $0 \notin Y$ et donc Y n'a pas de minimum. Nous disons que l'infimum de Y n'est pas atteint. ([8] : p8)

Définition 1.6 (Extremum/Optimum).

On nomme extremum d'une fonction f , tout point qui soit un minimum ou un maximum de f . Tandis que l'optimum (du latin *optimus* : meilleur) est considéré comme état, degré de développement de quelque chose jugé le plus favorable au regard des circonstances données.

Définition 1.7. Minimum local/Minimum global ([14] : pp32-33)

Considérons le problème d'optimisation suivant :

$$\min_{x \in \mathbb{R}} f(x) \tag{1.8}$$

sous contraintes

$$h(x) = 0, \quad (1.9)$$

$$g(x) \leq 0 \quad (1.10)$$

et

$$x \in X \quad (1.11)$$

Avec f une fonction de \mathbb{R}^n dans \mathbb{R} , h est une fonction de \mathbb{R}^n dans \mathbb{R}^m , g est une fonction de \mathbb{R}^n dans \mathbb{R}^p , et $X \subseteq \mathbb{R}^n$ est un ensemble convexe. Donc on a un problème d'optimisation à n variables de décision, m contraintes d'égalité et p contraintes d'inégalité.

Soit alors $Y = \{x \in \mathbb{R}^n / h(x) = 0, g(x) \leq 0 \text{ et } x \in X\}$ l'ensemble de vecteurs vérifiant toutes les contraintes. Considérons un vecteur $x^* \in Y$; nous présentons alors les définitions ci-après se rapportant à x^* :

1. Minimum local

x^* est un minimum local du problème (1.8)-(1.11) s'il existe $\varepsilon > 0$ tel que

$$f(x^*) \leq f(x), \forall x \in Y \text{ tel que } \|x - x^*\| < \varepsilon. \quad (1.12)$$

2. Minimum local strict

x^* est un minimum local strict du problème (1.8)-(1.11) s'il existe $\varepsilon > 0$ tel que

$$f(x^*) < f(x), \forall x \in Y, x \neq x^* \text{ tel que } \|x - x^*\| < \varepsilon. \quad (1.13)$$

3. Minimum global

x^* est un minimum global du problème (1.8)-(1.11) si

$$f(x^*) \leq f(x), \forall x \in Y. \quad (1.14)$$

4. Minimum global strict

x^* est un minimum global strict du problème (1.8)-(1.11) si

$$f(x^*) < f(x), \forall x \in Y, x \neq x^*. \quad (1.15)$$

Les notions de **maximum local**, **maximum local strict**, **maximum global** et **maximum global strict** se définissent de façon similaire en inversant le sens des inégalités dans les définitions ci-haut données.

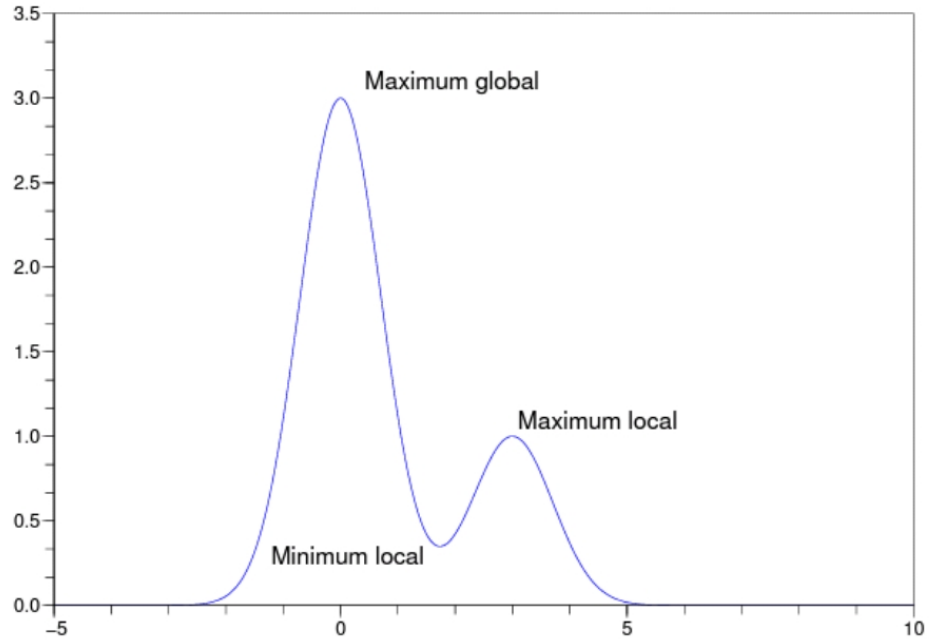


FIGURE 1.1 – Exemples de minima et maxima locaux et globaux pour la fonction

$$f : x \rightarrow 3e^{-x^2} + e^{-(x-3)^2}. \quad ([8] : \text{p14})$$

Définition 1.8. Point intérieur

Soit $Y \subset \mathbb{R}^n$ et $y \in Y$. On dit que y est intérieur à Y s'il existe un voisinage de y contenu dans Y . D'une manière équivalente, y est intérieur à Y s'il existe $\varepsilon > 0$ tel que

$$\forall z \in Y, \text{ on a } \|z - y\| \leq \varepsilon. \quad (1.16)$$

Définition 1.9. Point admissible

Soit le problème d'optimisation (1.8)-(1.11). Un point $x \in \mathbb{R}^n$ est dit admissible s'il vérifie toutes les contraintes (1.9)-(1.11). Néanmoins, dans la littérature ce concept est parfois appelé **solution admissible** ou **vecteur admissible**

Définition 1.10. (Direction admissible)

Soit le problème d'optimisation (1.8)-(1.11) et soit un point $x \in \mathbb{R}^n$ admissible. Une direction d sera dite admissible en x s'il existe $\eta > 0$ tel que $x + \alpha d$ soit admissible pour tout

$$0 < \alpha \leq \eta. \quad (1.17)$$

1.2.3 convexité et concavité

La convexité et la concavité jouent un rôle important en optimisation, en effet lorsque la fonction objectif n'est pas convexe, il est pratiquement impossible d'identifier un optimum global du problème (1.8)-(1.11). Ainsi, l'importance de la convexité est liée aux problèmes de minimisation alors que dans les problèmes de maximisation la concavité est nécessaire.

Définition 1.11. Ensemble convexe ([18] : p26)

Un ensemble $X \subseteq \mathbb{R}^n$ est dit convexe si pour tout $x \in X$ et pour tout $y \in X$, alors

$$\lambda x + (1 - \lambda)y \in X, \text{ pour tout } 0 \leq \lambda \leq 1. \quad (1.18)$$

En langage géométrique, cela signifie que si les extrémités d'un segment appartiennent à un ensemble X , il en est de même du segment tout entier.

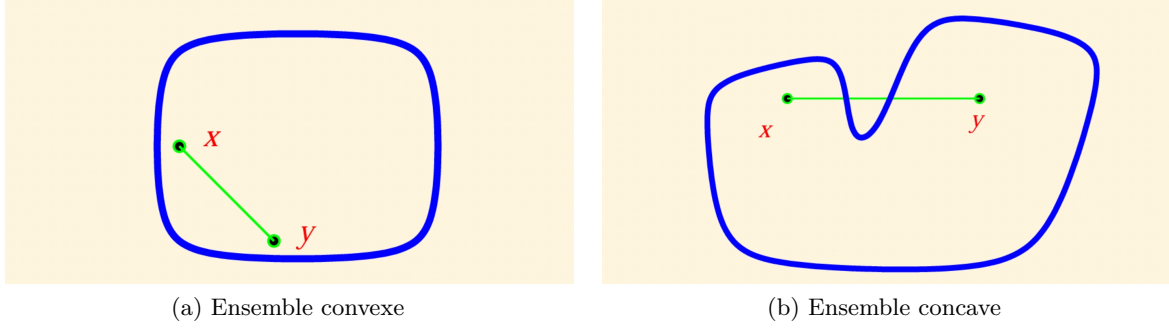


FIGURE 1.2 – Convexité et concavité

Définition 1.12. Fonction convexe/Fonction strictement convexe

– Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite convexe si $\forall x, y \in \mathbb{R}^n$ et $\forall \lambda \in [0, 1]$, on a :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (1.19)$$

Autrement dit, une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite convexe si, pour tout vecteur x et y de \mathbb{R}^n , le graphe de f entre x et y se trouve sous le segment de droite reliant $x, f(x)$ et $(y, f(y))$ dans \mathbb{R}^{n+1} . Par ailleurs,

– Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite strictement convexe si $\forall x, y \in \mathbb{R}^n$, $x \neq y$ et $\forall \lambda \in]0, 1[$, on a :

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y). \quad (1.20)$$

Définition 1.13. Fonction concave/Fonction strictement concave

– Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite concave si $-f$ est une fonction convexe; i.e $\forall x, y \in \mathbb{R}^n$ et $\forall \lambda \in [0, 1]$, on a :

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y). \quad (1.21)$$

De même,

– Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est strictement concave si

$$f(\lambda x + (1 - \lambda)y) > \lambda f(x) + (1 - \lambda)f(y). \quad (1.22)$$

Définition 1.14. Fonction fortement convexe ([17] : p11)

Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite fortement convexe s'il existe $\alpha > 0$ tel que

$$f(\lambda y + (1 - \lambda)x) \leq \lambda f(y) + (1 - \lambda)f(x) - \alpha \lambda(1 - \lambda) \|y - x\|^2, \forall x, y \in \mathbb{R}^n, \forall \lambda \in [0, 1]. \quad (1.23)$$

Toute fonction fortement convexe l'est strictement et simplement. La réciproque est fausse.

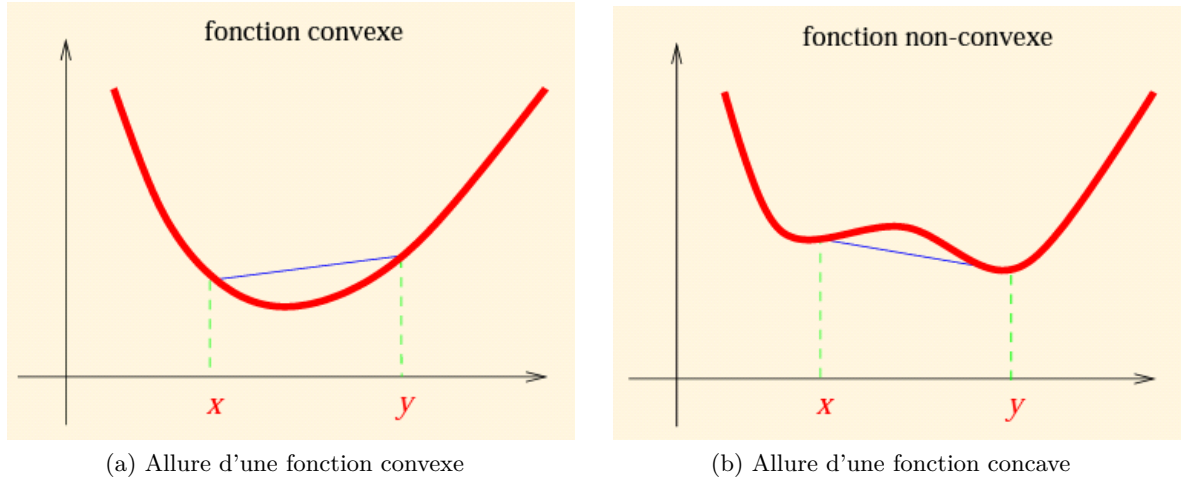


FIGURE 1.3 – Fonctions convexe et concave. ([18] : p28)

Néanmoins, la convexité et la concavité ne sont pas des propriétés complémentaires, une fonction peut n'être ni concave ni convexe.

1.2.4 Différentiabilité

Définition 1.15. Gradient ([14] : p44)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue, on appelle **gradient** de f ; la fonction notée $\nabla f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ et définie par

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}. \quad (1.24)$$

Définition 1.16. Dérivée directionnelle ([4] : p15)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue. Soient $x \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$. La dérivée directionnelle de f en x dans la direction d est

$$f'_d(x) = \lim_{\alpha \rightarrow 0^+} \frac{f(x + \alpha d) - f(x)}{\alpha} = \nabla f(x)^T d. \quad (1.25)$$

Définition 1.17. Fonction différentiable

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue, si pour tout $d \in \mathbb{R}^n$, la dérivée directionnelle de f dans la direction d existe alors la fonction f est dite différentiable. ([18] : p20)

Définition 1.18. Fonction coercive ([4] : p24 et [14] : p483)

Soit $X \subseteq \mathbb{R}^n$, et soit $f : X \rightarrow \mathbb{R}$ une fonction à valeurs réelles; f est dite coercive si pour toute suite $(x_k)_k$ d'éléments de X telle que $\|x_k\| \rightarrow \infty$ pour une norme quelconque, on a

$$\lim_{k \rightarrow \infty} f(x_k) = \infty. \quad (1.26)$$

Définition 1.19. Direction de descente ([4] : p44)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable, soient $x, d \in \mathbb{R}^n$. La direction d est une direction de descente en x si

$$d^T \nabla f(x) < 0. \quad (1.27)$$

Théorème 1.1. Plus forte pente ([14] : p47)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable, soient $x \in \mathbb{R}^n$, et $d^* = \nabla f(x)$. Alors, pour tout $d \in \mathbb{R}^n$ tel que $\|d\| = \|\nabla f(x)\|$ on a

$$d^T \nabla f(x) \leq d^{*T} \nabla f(x) = \nabla f(x)^T \nabla f(x). \quad (1.28)$$

Preuve. Soit d une direction quelconque, nous avons :

$$\begin{aligned} d^T \nabla f(x) &\leq \|d\| \|\nabla f(x)\| \quad (\text{Cauchy - Schwartz}) \\ &= \|\nabla f(x)\|^2 \quad (\text{par hypothèse } \|d\| = \|\nabla f(x)\|) \\ &= \nabla f(x)^T \nabla f(x) \quad (\text{par déf. du produit scalaire}) \\ &= d^{*T} \nabla f(x) \quad (\text{par déf. de } d^*). \end{aligned}$$

Comme $d^T \nabla f(x) = \|\nabla f(x)\|^2 \geq 0$, la fonction est non décroissante dans la direction d^* , qui correspond bien à la plus forte pente. Si la direction du gradient correspond à la plus forte pente de la fonction en x , il suffit de considérer la direction opposée au gradient $-\nabla f(x)$ pour trouver la plus forte pente. \square

Corollaire 1.1. Plus forte descente ([5] : p28)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable, soient $x \in \mathbb{R}^n$, et $d^* = -\nabla f(x)$. Alors, pour tout $d \in \mathbb{R}^n$ tel que $\|d\| = \|\nabla f(x)\|$ on a :

$$-\nabla f(x)^T \nabla f(x) = d^{*T} \nabla f(x) \leq d^T \nabla f(x) \quad (1.29)$$

et la direction opposée au gradient est celle dans laquelle la fonction a la plus forte descente.

Preuve. Soit d une direction quelconque, nous avons :

$$\begin{aligned} -d^T \nabla f(x) &\leq \nabla f(x)^T \nabla f(x) \quad (\text{en appliquant la plus forte pente à } -d) \\ &= -d^{*T} \nabla f(x) \quad (\text{par déf. de } d^*) \\ \text{et } -d^T \nabla f(x) &\leq -(d^*)^T \nabla f(x) \end{aligned}$$

On obtient (1.29) en multipliant cette dernière égalité par -1. Comme $d^{*T} \nabla f(x) \leq 0$, la fonction est décroissante dans la direction d^* , qui correspond bien à la plus forte descente. \square

Définition 1.20. Matrice gradient ([14] : p52)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ telle que $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable, pour $i = 1, \dots, m$. Dans ce cas, f est différentiable. On appelle matrice gradient, la fonction notée $\nabla f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ et définie par

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}. \quad (1.30)$$

La matrice gradient est souvent utilisée dans sa forme transposée, et est alors appelée **la matrice Jacobienne de f** . On note $J(x) = \nabla f(x)^T$.

Définition 1.21. Matrice hessienne ([17] : p8)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable. On appelle matrice hessienne ou hessien de f la fonction notée $\nabla^2 f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ et définie par

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}. \quad (1.31)$$

La matrice hessienne est toujours symétrique ; c'est à dire $(\nabla^2 f(x) = \nabla^2 f(x)^T)$.

Par ailleurs, il est alors à rappeler que l'écriture $\frac{\partial^2 f}{\partial x_i \partial x_j}$ signifie que l'on a dérivé d'abord f par rapport à la j^e variable x_j puis par rapport à la i^e variable x_i . ([14] : p53)

Remarque 1.1. Cette remarque vient compléter la notion de convexité définie ci-haut pour une fonction donnée.

1. **Convexité par le gradient** : Soit $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable sur un ensemble convexe ouvert X . Une fonction f est convexe si et seulement si

$$f(y) - f(x) \geq (y - x)^T \nabla f(x), \forall x, y \in X. \quad (1.32)$$

f est strictement convexe si et seulement si

$$f(y) - f(x) > (y - x)^T \nabla f(x), \forall x, y \in X. \quad (1.33)$$

2. **Convexité par le hessien** : Soit $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable sur un ensemble convexe ouvert X . Si $\nabla^2 f(x)$ est semi-définie positive (resp. définie positive) pour tout x dans X , alors f est convexe (resp. strictement convexe).

Définition 1.22. Courbure ([14] : p54)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable. Soient $x, d \in \mathbb{R}^n$. On appelle courbure de la fonction f en x dans la direction d , la quantité

$$\frac{d^T \nabla^2 f(x) d}{d^T d}. \quad (1.34)$$

1.2.5 Signe d'une matrice

La matrice carrée $A \in \mathbb{R}^{n \times n}$ est dite :

- **Semi-définie positive** si $\forall x \in \mathbb{R}^n$ on a $x^T A x \geq 0$.
- **Définie positive** si $\forall x \in \mathbb{R}^n, x \neq 0$ on a $x^T A x > 0$.
- **Semi-définie négative** si $\forall x \in \mathbb{R}^n$ on a $x^T A x \leq 0$.
- **Définie négative** si $\forall x \in \mathbb{R}^n, x \neq 0$ on a $x^T A x < 0$. ([16] : p7)

Exemple 1.2. *Considérons une fonction*

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}; x \rightarrow f(x) = \frac{1}{2}x_1^2 + 2x_2^2. \quad (1.35)$$

1. *Le gradient de la fonction (1.35) est :*

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} x_1 \\ 4x_2 \end{pmatrix}.$$

2. *Le hessien de la fonction (1.35) est :*

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}.$$

3. *En considérant toujours la fonction (1.35) avec un point $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ et trois directions $d_1 = \nabla f(x) = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$, $d_2 = \begin{pmatrix} -1 \\ 3 \end{pmatrix}$ et $d_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$; calculons respectivement les dérivées directionnelles .
En effet ;*

$$d_1^T \nabla f(x) = \begin{pmatrix} 1 \\ 4 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = 1 + 16 = 17,$$

$$d_2^T \nabla f(x) = \begin{pmatrix} -1 \\ 3 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = -1 + 12 = 11 \text{ et}$$

$$d_3^T \nabla f(x) = \begin{pmatrix} -1 \\ -1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = -1 - 4 = -5.$$

4. *Considérons (1.35) et montrons que les directions $d_1 = -\nabla f(x) = \begin{pmatrix} -1 \\ -4 \end{pmatrix}$,*

$d_2 = \begin{pmatrix} 1 \\ -3 \end{pmatrix}$ et $d_3 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ sont des directions de descente au point $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$

En effet :

$$d_1^T \nabla f(x) = \begin{pmatrix} -1 \\ -4 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 & -4 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = -1 - 16 = -17 < 0,$$

$$d_2^T \nabla f(x) = \begin{pmatrix} 1 \\ -3 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & -3 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = 1 - 12 = -11 < 0 \text{ et}$$

$$d_3^T \nabla f(x) = \begin{pmatrix} -1 \\ -1 \end{pmatrix}^T \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix} = -1 - 4 = -5 < 0.$$

Il en résulte que d_1 , d_2 et d_3 sont des directions de descente car elles vérifient l'équation (1.27) de la définition 1.19.

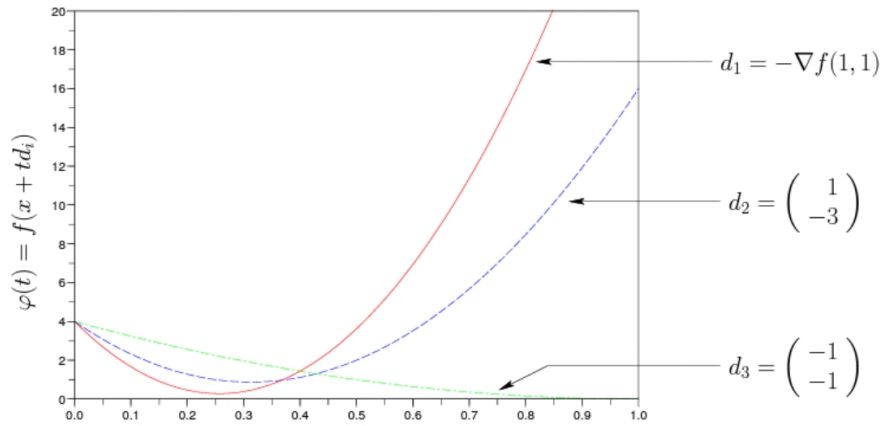


FIGURE 1.4 – Allure de la fonction $f(x) = \frac{1}{2}x_1^2 + 2x_2^2$ au point $x = (1 \ 1)^T$ dans trois directions données. ([8] : p26)

5. Trouvons les différentes courbures pour les deux directions $d_1 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$ et $d_2 = \begin{pmatrix} -1 \\ 3 \end{pmatrix}$ ci-haut données en considérant toujours la même valeur de x .
En effet :

$$\begin{aligned} \bullet \frac{d_1^T \nabla^2 f(x) d_1}{d_1^T d_1} &= \frac{\begin{pmatrix} 1 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix}}{\begin{pmatrix} 1 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix}} = \frac{\begin{pmatrix} 1 & 16 \end{pmatrix} \begin{pmatrix} 1 \\ 4 \end{pmatrix}}{17} = \frac{65}{17} = 3.8235. \\ \bullet \frac{d_2^T \nabla^2 f(x) d_2}{d_2^T d_2} &= \frac{\begin{pmatrix} -1 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} -1 \\ 3 \end{pmatrix}}{\begin{pmatrix} -1 & 3 \end{pmatrix} \begin{pmatrix} -1 \\ 3 \end{pmatrix}} = \frac{\begin{pmatrix} -1 & 12 \end{pmatrix} \begin{pmatrix} -1 \\ 3 \end{pmatrix}}{10} = \frac{37}{10} = 3.7. \end{aligned}$$

6. Les fonctions $f(x) = \frac{1}{2}x_1^2 + 2x_2^2$, $f(x) = x^4$, $f(x, y) = 2x^2 + 2xy + y^2 + 10x + 10y$ et $f(x, y) = x^2 + y^2$ sont strictement convexes (en vertu de la convexité par le hessien cfr. remarque 1.1 à la page 14 car chacune d'elles admet un hessien défini positif).

1.3 Conditions d'Optimalité

([14] : pp125-179, [7] : pp28-45 et [4] : pp9-26)

Les conditions d'optimalité jouent trois grands rôles dans le développement des algorithmes :

1. Elles procurent une analyse théorique du problème.
2. Elles inspirent directement des idées des algorithmes .

3. Elles permettent de déterminer un critère d'arrêt pour les algorithmes itératifs .

Dans cette section, nous allons les présenter en deux aspects ; d'une part celles de l'optimisation sans contrainte et d'autre part celles de l'optimisation avec contraintes .

1.3.1 Optimisation sans contrainte

Théorème 1.2. Condition nécessaire d'optimalité

Soit x^* un minimum local d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Si f est différentiable dans un voisinage ouvert V de x^* alors :

1.

$$\nabla f(x^*) = 0. \quad (1.36)$$

Si de plus, f est deux fois différentiable alors :

2.

$$\nabla^2 f(x^*) \text{ est sémi-définie positive.} \quad (1.37)$$

La condition (1.36) est appelée condition nécessaire d'optimalité du premier ordre et (1.37) est celle du deuxième ordre .

Preuve. D'après le développement de Taylor à l'ordre 1 en x^* on a :

$$f(x) = f(x^*) + (x - x^*)^T \nabla f(x^*) + \|x - x^*\| \sum(x) \quad (1.38)$$

où $\sum(x)$ une fonction qui tend vers 0 quand x tend x^* . En particulier, en choisissant $x = x^* - \theta \nabla f(x^*)$ avec $\theta \in \mathbb{R}$, on obtient :

$$\begin{aligned} f(x) - f(x^*) &= -\theta \|\nabla f(x^*)\|^2 + \theta \varepsilon_1(\theta) \\ &= \theta(-\|\nabla f(x^*)\|^2 + \varepsilon_1(\theta)) \end{aligned} \quad (1.39)$$

où $\varepsilon_1(\theta)$ est une fonction qui tend vers 0 quand θ tend vers 0. Pour θ positif, $f(x) - f(x^*)$ est du signe de $-\|\nabla f(x^*)\|^2 + \theta \varepsilon_1(\theta)$. Si $\nabla f(x^*) \neq 0$, il existe dans tout voisinage de x^* des points x vérifiant $f(x) \leq f(x^*)$ (pour θ petit, $f(x) - f(x^*)$ est du signe de $-\|\nabla f(x^*)\|^2$, si on suppose ce terme non nul) contradiction avec l'optimalité locale de x^* . D'où le résultat (1.36). Supposons maintenant qu'il existe $h \in \mathbb{R}^1$ tel qu'on ait $h^t \nabla^2 f(x^*) h < 0$. On a alors d'après le développement de Taylor à l'ordre 2

$$f(x^* + \theta h) - f(x^*) = \theta^2 \left(\frac{1}{2} h^t \nabla^2 f(x^*) h \right) + \varepsilon_2(\theta) \quad (1.40)$$

Où $\varepsilon_2(\theta)$ est une fonction qui tend vers 0 quand θ tend vers 0. Pour θ assez petit, la différence $f(x^* + \theta h) - f(x^*)$ serait négative. Ce qui contredit l'hypothèse sur x^* . \square

La condition nécessaire du premier ordre joue un rôle central en optimisation, les vecteurs x qui vérifient cette condition sont appelés **Points critiques** ou **Points stationnaires**. Parmi eux, il y a des minima locaux et des maxima locaux et éventuellement des points qui ne sont ni l'un ni l'autre et sont appelés **Points selle** aux quels on peut ajouter l'existence des **Points réguliers**. Il est à signaler que lorsqu'on est simplement en \mathbb{R} ; les points selles sont des points d'inflexion.

Définition 1.23. Point critique

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Tout vecteur $x \in \mathbb{R}^n$ tel que $\nabla f(x) = 0$ est appelé point critique ou point stationnaire de f .

Théorème 1.3. Condition suffisante d'optimalité

Soit une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable dans un sous-ensemble ouvert V de \mathbb{R} et soit $x^* \in V$ vérifiant les conditions

1.

$$\nabla f(x^*) = 0 \quad (1.41)$$

et

2.

$$\nabla^2 f(x^*) \text{ est définie positive} \quad (1.42)$$

Alors x^* est un minimum local de f .

Preuve. La matrice $\nabla^2 f(x^*)$ étant définie positive, il existe $\alpha > 0$ tel que

$$\forall h \in \mathbb{R}^n, h^t \nabla^2 f(x^*) h \geq \alpha \|h\|^2$$

En effet, plaçons-nous sur la sphère S de centre D et de rayon 1 et définissons α par $\alpha = \inf(h^t \nabla^2 f(x^*) h)$.

La sphère étant un compact, α est atteint : $\exists h_o \in \dots$ tel que $\alpha = \inf(h_o^t \nabla^2 f(x^*) h_o) > 0$. On déduit aisément la proposition précédente.

D'où en appliquant la formule de Taylor avec $h = x - x^*$ on a

$$f(x) - f(x^*) = f(x^* + h) - f(x^*) \geq \|h\|^2 \left(\frac{\alpha}{2} + \varepsilon(h) \right) \quad (1.43)$$

où $\varepsilon(h)$ est une fonction qui tend vers 0 quand h tend vers 0. Ce qui montre le théorème car pour des normes assez petites de h , $\frac{\alpha}{2} + \varepsilon(h)$ est du signe de α , i.e positif. \square

Théorème 1.4. Condition suffisante d'optimalité globale

Soit une fonction continue $f : \mathbb{R}^n \rightarrow \mathbb{R}$, et soit $x^* \in \mathbb{R}^n$ un minimum local de f . Si f est une fonction convexe, alors x^* est un minimum global de f . Si de plus f est strictement convexe, x^* est l'unique minimum global de f .

Preuve. Supposons par l'absurde qu'il existe un autre minimum local $x^+ \neq x^*$ tel que $f(x^+) < f(x^*)$. Par convexité de f , on a $f(\alpha x^* + (1 - \alpha)x^+) \leq \alpha f(x^*) + (1 - \alpha)f(x^+)$ où $\alpha \in [0, 1]$.

Comme $f(x^+) < f(x^*)$, nous avons pour chaque $\alpha \in]0, 1]$

$$f(\alpha x^* + (1 - \alpha)x^+) < \alpha f(x^*) + (1 - \alpha)f(x^+) = f(x^*) \quad (1.44)$$

Considérons $\varepsilon > 0$ arbitraire, et montrons que l'équation (1.12) est contredite.

Si $\varepsilon \geq \|x^* - x^+\|$, (1.12) n'est pas vérifiée pour $x = x^+$, en prenant $\alpha = 1$ dans (1.42).

Si $\varepsilon < \|x^* - x^+\|$, considérons $0 < \eta < 1$ tel que $\|\eta x^* + (1 - \eta)x^+\| = \varepsilon$, dans ce cas (1.12) n'est pas vérifiée pour $x = \alpha x^* + (1 - \alpha)x^+$ avec $\eta \leq \alpha < 1$. Par (1.42) de tels α existent toujours.

Considérons maintenant une fonction strictement convexe, et supposons que x^* et y^* soient des minima globaux distincts, et donc $x^* \neq y^*$ et $f(x^*) = f(y^*)$ et donc on a $f(\alpha x^* + (1 - \alpha)y^*) < \alpha f(x^*) + (1 - \alpha)f(y^*) = f(x^*) = f(y^*)$, $\forall \alpha \in]0, 1[$, ce qui contredit que x^* et y^* sont des minima globaux. \square

Théorème 1.5. Conditions d'optimalité pour les problèmes quadratiques

Considérons le problème

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + g^T x + c \quad (1.45)$$

où Q est une matrice symétrique $n \times n$, $g \in \mathbb{R}^n$ et $c \in \mathbb{R}$.

1. Si Q n'est pas semi-définie positive, alors le problème (1.45) ne possède pas de solution, c'est-à-dire qu'il n'existe aucun $x \in \mathbb{R}^n$ qui soit un minimum local de (1.45).
2. Si Q est définie positive, alors

$$x^* = -Q^{-1}g \quad (1.46)$$

est l'unique minimum global de (1.45).

Preuve . Nous avons que $\nabla f(x) = Qx + g$ et $\nabla^2 f(x) = Q$.

1. Supposons par l'absurde qu'il existe x^* un minimum local de (1.45), par (1.42),

$$\nabla^2 f(x) = Q$$

est semi-définie positive, ce qui contredit l'hypothèse .

2. Si Q est définie positive, le point x^* dans l'équation (1.46) est bien défini et $\nabla f(x^*) = -QQ^{-1}g + g = 0$, les conditions d'optimalité (1.41) et (1.42) sont vérifiées et x^* est un minimum local de f . De plus f est strictement convexe, alors x^* est l'unique minimum global.

□

Exemple 1.3. Le problème du Rosenbrock ([14] : p317)

$$\min_{x \in \mathbb{R}^2} f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (1.47)$$

admet $x^* = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ comme minimum local.

En effet,

$$\begin{aligned} \bullet f(x) &= 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ &= 100(x_2^2 - 2x_2x_1^2 + x_1^4) + 1 - 2x_1 + x_1^2 \\ &= 100x_2^2 - 200x_2x_1^2 + 100x_1^4 + 1 - 2x_1 + x_1^2 \\ &= 100x_1^4 - (200x_2 - 1)x_1^2 - 2x_1 + 100x_2^2 + 1 \\ \bullet \nabla f(x) &= \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 400x_1^3 - 2(200x_2 - 1)x_1 - 2 \\ 200x_2 - 200x_1^2 \end{pmatrix} \\ \bullet \nabla^2 f(x) &= \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix} \\ \bullet &\text{ Cherchons les points critiques ; En effet,} \end{aligned}$$

$$\bullet \nabla f(x) = 0 \Leftrightarrow \begin{cases} 400x_1^3 - 2(200x_2 - 1)x_1 - 2 = 0 & (1) \\ 200x_2 - 200x_1^2 = 0 & (2) \end{cases}$$

• De (2) : $x_2 = x_1^2$ (3), en portant (3) dans (1) on trouve :

$$400x_1^3 - 400x_1^3 + 2x_1 - 2 = 0 \Leftrightarrow x_1 = 1 \quad (4)$$

Et en portant (4) dans (3) on trouve finalement $x_2 = 1$

D'où $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ est un point critique

• Montrons enfin que $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ est un minimum local de (1.47).

D'une part $\nabla f(1 \ 1) = \begin{pmatrix} 0 & 0 \end{pmatrix} = 0$; ce qui prouve que l'équation (1.36) à la page 17 reste vérifiée. Et d'autre part, le hessien en x est défini positif :

$$\begin{pmatrix} -400 + 2 + 1200 & -400 \\ -400 & 200 \end{pmatrix} = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}.$$

$$\begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 402 & -200 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 402 - 200 = 202 > 0 \quad (\text{positif}).$$

Par suite, x vérifie les conditions d'optimalité. Il en résulte que $x = \begin{pmatrix} 1 & 1 \end{pmatrix}^T$ est un minimum local de (1.47)

1.3.2 Optimisation avec contraintes

Le développement des conditions d'optimalité avec contraintes est basé sur la même intuition que dans le cas sans contraintes, il n'y a pas moyen de descendre à partir d'un minimum. Ce pendant, on ne peut plus appliquer les conditions vues dans la sous section (1.3.1).

Théorème 1.6. Conditions d'optimalité pour contraintes générales

Soit x^* un minimum local du problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ sous contraintes $h(x) = 0$, $g(x) \leq 0$ et $x \in X$. Alors

$$\nabla f(x^*)^T d \geq 0 \quad (1.48)$$

pour toute direction d admissible à la limite en x^* .

Preuve . Supposons par l'absurde qu'il existe une direction admissible à la limite d telle que $\nabla f(x^*)^T d < 0$ et considérons la suite admissible $(x_k)_{k \in \mathbb{N}}$. Par le théorème de Taylor avec $d = x_k - x^*$ on a :

$$f(x_k) = f(x^*) + (x_k - x^*)^T \nabla f(x^*) + \theta(\|x_k - x^*\|). \quad (1.49)$$

Comme $d = \lim_k (x_k - x^*)$ et que $\nabla f(x^*)^T d < 0$, il existe un indice k tel que $(x_k - x^*)^T \nabla f(x^*) < 0$ pour tout $k \in K$. De plus, le terme $\theta(\|x_k - x^*\|)$ peut être rendu aussi petit que l'on veut en rendant k suffisamment grand. Dès lors, il existe k suffisamment grand tel que $f(x_k) < f(x^*)$, ce qui contredit l'optimalité de x^* . \square

Ce résultat général ne tenant pas compte d'une structure éventuelle dans les contraintes, voici des conditions d'optimalité pour des problèmes particuliers.

1. Contraintes convexes I

1. Condition nécessaire

Soit x^* un minimum local du problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable sur X et X est un ensemble convexe non vide. Alors $\forall x \in X$,

$$\nabla f(x^*)^T (x - x^*) \geq 0 \quad (1.50)$$

2. Condition suffisante

Soit le problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ où X est un ensemble convexe fermé non vide, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable et convexe sur X . Alors (1.50) est une condition suffisante pour que x^* soit un minimum global de f sur X .

2. Contraintes convexes II

Soit le problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ où X est un ensemble convexe fermé non vide, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable. Si x^* est un minimum local, alors

$$x^* = [x^* - \alpha \nabla f(x^*)]^P, \forall \alpha > 0. \quad (1.51)$$

Si de plus f est convexe, (1.51) est suffisante pour que x^* optimise f sur X .

3. Multiplicateurs de Lagrange

a. Conditions nécessaires

Les conditions nécessaires d'optimalité sont généralement appelées *Conditions de KARUSH-KUHN-TUCKER (KKT)* et utilisent une fonction spécifique appelée fonction lagrangienne.

Définition 1.24. Fonction lagrangienne

Soit le problème d'optimisation (1.8)-(1.10) $\min f(x)$ sous contraintes $h(x) = 0$ et $g(x) \leq 0$. Et soient les vecteurs $\lambda \in \mathbb{R}^m$ et $\mu \in \mathbb{R}^P$. La fonction $L : \mathbb{R}^{n+m+p} \rightarrow \mathbb{R}$ définie par

$$\begin{aligned} L(x, \lambda, \mu) &= f(x) + \lambda^T h(x) + \mu^T g(x) \\ &= f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^p \mu_j g_j(x) \end{aligned} \quad (1.52)$$

est appelée lagrangien ou fonction lagrangienne du problème (1.8)-(1.10).

a.1 Contraintes linéaires

Soit x^* un minimum local du problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ sous contraintes

$$Ax = b \quad (1.53)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable, et $A \in \mathbb{R}^{m \times n}$ est de rang plein. Alors, il existe un vecteur unique $\lambda^* \in \mathbb{R}^m$ tel que

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= \nabla f(x^*) + A^T \lambda^* \\ &= 0 \end{aligned} \quad (1.54)$$

où L est la fonction lagrangienne. Si f est deux fois différentiable, alors

$$Y^T \nabla_{xx}^2(x^*, \lambda^*) Y \geq 0, \forall Y \in \mathbb{R}^n \text{ tel que } AY = 0 \quad (1.55)$$

a.2 Contraintes d'égalité

Soit x^* un minimum local du problème $\min_{x \in \mathbb{R}^n}$ sous contrainte

$$h(x) = 0 \quad (1.56)$$

avec $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable et $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ continûment différentiable. Si les contraintes sont linéairement indépendantes, alors il existe un vecteur unique λ^* tel que

$$\nabla L(x^*, \lambda^*) = 0 \quad (1.57)$$

où L est la fonction lagrangienne. Si f et h sont deux fois différentiable, alors

$$Y^T \nabla_{xx}^2(x^*, \lambda^*) Y \geq 0; \forall Y \in D_* \quad (1.58)$$

où D_* est le cône des directions en x^* , le vecteur $\lambda^* = [\lambda_1 \dots \lambda_p]^T$ est appelé *Multiplicateur de Lagrange* et est déterminé de façon unique :

$$\lambda^* = - \left(\nabla h(x^*)^T \nabla h(x^*)^{-1} \nabla h(x^*)^T \nabla f(x^*) \right). \quad (1.59)$$

b. Conditions suffisantes

La démonstration de la condition suffisante d'optimalité utilise une fonction appelée **Lagrangien augmenté**, qui sera également utilisé dans le cadre de certains algorithmes .

Définition 1.25. Lagrangien augmenté

Soit le problème d'optimisation avec contrainte d'égalité (1.8)-(1.9) $\min_{x \in \mathbb{R}^n} f(x)$ sous contrainte $h(x) = 0$ et soit un paramètre $c \in \mathbb{R}, c > 0$, appelé paramètre de pénalité. La fonction lagrangienne du problème

$$\min_{x \in \mathbb{R}^n} f(x) + \frac{c}{2} \| h(x) \|^2 \quad (1.60)$$

sous contrainte

$$h(x) = 0 \quad (1.61)$$

est appelé le lagrangien augmenté du problème (1.8)-(1.9) et s'écrit

$$\begin{aligned} L_c(x, \lambda) &= L(x, \lambda) + \frac{c}{2} \| h(x) \|^2 \\ &= f(x) + \lambda^T h(x) + \frac{c}{2} \| h(x) \|^2. \end{aligned} \quad (1.62)$$

L'idée de la fonction objectif (1.60) est de pénaliser au niveau de la fonction objectif des points x qui violent les contraintes, d'où le nom de paramètre de pénalité pour c .

Contraintes d'égalité

Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ des fonctions deux fois différentiables. Soient $x^* \in \mathbb{R}^n$ et $\lambda^* \in \mathbb{R}^m$ tels que

$$\nabla L(x^*, \lambda^*) = 0 \quad (1.63)$$

et

$$Y^T \nabla_{xx}^2 L(x^*, \lambda^*) Y > 0, \forall Y \in D(x^*), Y \neq 0 \quad (1.64)$$

où L est la fonction lagrangienne du problème d'optimisation $\min_{x \in \mathbb{R}^n} f(x)$ sous contrainte $h(x) = 0$, et $D(x^*)$ le cône des directions en x^* . Alors x^* est un minimum local du problème d'optimisation .

4. Problèmes quadratiques

Théorème 1.7. Solution analytique d'un problème d'optimisation

Soit le problème quadratique

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x \quad (1.65)$$

Sous contrainte

$$Ax = b \quad (1.66)$$

avec A de rang plein. Si la matrice Q est définie positive alors le système linéaire

$$\begin{pmatrix} Q & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix} \quad (1.67)$$

est non singulier et possède une solution (x^*, λ^*) unique donnée par

$$x^* = Q^{-1}(A^T \lambda^* - c) \quad (1.68)$$

et

$$\lambda^* = (AQ^{-1}A^T)^{-1} (AQ^{-1}c + b).$$

Preuve . Soit $Z \in \mathbb{R}^{n \times (n-m)}$ une matrice dont les colonnes forment une base de l'espace nul de A . C'est à dire telle que $AZ = 0$. Comme Q est définie positive, alors $Z^T Q Z$ également et le lemme y relatif s'applique pour montrer la non singularité du système et l'unicité de la solution. Soit L une matrice triangulaire inférieure telle que $Q = LL^T$ et posons $Y = L^T X$. Le problème (1.68)-(1.69) s'écrit alors

$$\min \frac{1}{2} Y^T L^{-1} L L^T L^{-T} Y + c^T L^{-T} Y = \frac{1}{2} Y^T Y + c^T L^{-T} Y \quad (1.69)$$

Sous contrainte

$$AL^{-T} Y = b. \quad (1.70)$$

La solution du problème est donnée par le même lemme, en remplaçant c par $L^{-1}c$ et A par AL^{-T} . Ainsi,

$$\begin{aligned} \lambda^* &= (AL^{-T}L^{-1}A^T)^{-1} (AL^{-T}L^{-1}c + b) \\ &= (AQ^{-1}A^T)^{-1} (AQ^{-1}c + b) \end{aligned} \quad (1.71)$$

et $Y^* = L^{-1}A^T \lambda^* - L^{-1}c$. Il reste à poser $Y^* = L^T x^*$ pour obtenir le résultat. Ainsi,

$$\begin{aligned} Y^* &= L^T x^* \\ \Leftrightarrow L^T x^* &= L^{-1}A^T \lambda^* - L^{-1}c \\ \Leftrightarrow x^* &= L^{-1}L^{-T}(A^T \lambda^* - c) \\ \Leftrightarrow x^* &= (LL^T)^{-1}(A^T \lambda^* - c) \\ \Leftrightarrow x^* &= Q^{-1}(A^T \lambda^* - c) \end{aligned}$$

D'où le résultat (1.68) . □

Exemple 1.4. Exemples d'illustration

1. Considérons le problème

$$\min_{x \in \mathbb{R}^2} f(x) = 3x_1^2 + x_2^2 \quad (1.72)$$

sous contrainte

$$2x_1 + x_2 = 1. \quad (1.73)$$

En effet :

• $\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 6x_1 \\ 2x_2 \end{pmatrix}$ et de la contrainte (1.73); en déduisons la forme de

l'équation (1.53) à la page 21.

$$2x_1 + x_2 \Rightarrow \overbrace{\begin{pmatrix} 2 & 1 \end{pmatrix}}^A \overbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}^x = \overbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}^b \quad (\text{car la contrainte est linéaire.})$$

• $L(x, \lambda) = 3x_1^2 + x_2^2 + (2x_1 + x_2 - 1)\lambda$ (est le langrangien du problème (1.72)-(1.73))

$$\bullet \nabla L(x, \lambda) = \begin{pmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \frac{\partial L}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} 6x_1 + 2\lambda \\ 2x_2 + \lambda \\ 2x_1 + x_2 - 1 \end{pmatrix}$$

$$\bullet \nabla L(x, \lambda) = 0 \Leftrightarrow \begin{cases} 6x_1 + 2\lambda = 0 & (1) \\ 2x_2 + \lambda = 0 & (2) \\ 2x_1 + x_2 - 1 = 0 & (3) \end{cases}$$

$$\text{De (1) : } \lambda = -3x_1 \quad (4), \text{ portons (4) dans (2) : } 2x_2 - 3x_1 = 0 \Leftrightarrow x_1 = \frac{2}{3}x_2 \quad (5)$$

$$\text{Portons (5) dans (3) : } \frac{4}{3}x_2 + x_2 = 1, \Leftrightarrow 7x_2 = 3 \Leftrightarrow x_2 = \frac{3}{7} \quad (6)$$

$$\text{Par ailleurs, en portant (6) dans (5), on trouve } x_1 = \frac{2}{7} \quad (7), \text{ portons enfin (7)}$$

$$\text{dans (4) : } \lambda = -\frac{6}{7},$$

$$D'où $x^* = (x_1, x_2) = (\frac{2}{7}, \frac{3}{7})$ et $\lambda^* = -\frac{6}{7}$.$$

• Vérifions les conditions d'optimalité :

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= \nabla f(x^*) + A^T \lambda^* \\ &= \begin{pmatrix} 6 \times \frac{2}{7} \\ 2 \times \frac{3}{7} \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \end{pmatrix} \times \frac{-6}{7} \\ &= \begin{pmatrix} \frac{12}{7} \\ \frac{6}{7} \end{pmatrix} + \begin{pmatrix} -\frac{12}{7} \\ -\frac{6}{7} \end{pmatrix} \\ &= \begin{pmatrix} \frac{12}{7} - \frac{12}{7} \\ \frac{6}{7} - \frac{6}{7} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &= 0. \end{aligned}$$

En vertu de tout ce qui précède, il en découle que le problème (1.72) - (1.73) admet un minimum en $x^* = (\frac{2}{7}, \frac{3}{7})$ avec $\lambda^* = -\frac{6}{7}$ vérifiant les conditions d'optimalité (1.54) et (1.55).

2. Considérons le problème

$$\min_{x \in \mathbb{R}^4} f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 \quad (1.74)$$

sous contraintes

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 - x_2 + x_4 = 1 \end{cases} \quad (1.75)$$

Cherchons le minimum du problème considéré.

En effet ;

$$\bullet \nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \frac{\partial f}{\partial x_3} \\ \frac{\partial f}{\partial x_4} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \\ 2x_4 \end{pmatrix}.$$

$$\bullet \begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 - x_2 + x_4 = 1 \end{cases} \Rightarrow \overbrace{\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}}^A \overbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}}^x = \overbrace{\begin{pmatrix} 1 \\ 1 \end{pmatrix}}^b$$

(car les contraintes sont linéaires en vertu de (1.53)).

• Le lagrangien du problème (1.74)-(1.75) est :

$$L(x, \lambda) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + (x_1 + x_2 + x_3 - 1)\lambda_1 + (x_1 - x_2 + x_4 - 1)\lambda_2$$

$$\bullet \nabla L(x, \lambda) = \begin{pmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \frac{\partial L}{\partial x_3} \\ \frac{\partial L}{\partial x_4} \end{pmatrix} = \begin{pmatrix} 2x_1 + \lambda_1 + \lambda_2 \\ 2x_2 + \lambda_1 - \lambda_2 \\ 2x_3 + \lambda_1 \\ 2x_4 + \lambda_2 \\ x_1 + x_2 + x_3 - 1 \\ x_1 - x_2 + x_4 - 1 \end{pmatrix}$$

$$\bullet \nabla L(x, \lambda) = 0 \Leftrightarrow \begin{cases} 2x_1 + \lambda_1 + \lambda_2 = 0 & (1) \\ 2x_2 + \lambda_1 - \lambda_2 = 0 & (2) \\ 2x_3 + \lambda_1 = 0 & (3) \\ 2x_4 + \lambda_2 = 0 & (4) \\ x_1 + x_2 + x_3 - 1 = 0 & (5) \\ x_1 - x_2 + x_4 - 1 = 0 & (6) \end{cases}$$

De (3) : $\lambda_1 = -2x_3$ (7) et de (4) : $\lambda_2 = -2x_4$ (8). En portant (7) et (8) dans le système ci-haut formé on trouve :

$$\begin{cases}
2x_1 - 2x_3 - 2x_4 = 0 & (9) \\
2x_2 - 2x_3 + 2x_4 = 0 & (10) \\
x_1 + x_2 + x_3 = 1 & (11) \\
x_1 - x_2 + x_4 = 1 & (12)
\end{cases}$$

$$\Leftrightarrow \begin{cases}
2x_1 + 0x_2 - 2x_3 - 2x_4 = 0 \\
0x_1 + 2x_2 - 2x_3 + 2x_4 = 0 \\
x_1 + x_2 + x_3 + 0x_4 = 1 \\
x_1 - x_2 + 0x_3 + x_4 = 1
\end{cases}$$

$$\Leftrightarrow \begin{pmatrix} 2 & 0 & -2 & -2 \\ 0 & 2 & -2 & 2 \\ 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$\Leftrightarrow \begin{matrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{matrix} \left(\begin{array}{cccc|c} 2 & 0 & -2 & -2 & 0 \\ 0 & 2 & -2 & 2 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & -1 & 0 & 1 & 1 \end{array} \right)$$

$$\Leftrightarrow \begin{matrix} L'_1 \\ L'_2 \\ L'_3 \\ L'_4 \end{matrix} = \begin{matrix} L_1 \\ L_2 \\ L_1 - 2L_3 \\ L_3 - L_4 \end{matrix} \left(\begin{array}{cccc|c} 2 & 0 & -2 & -2 & 0 \\ 0 & 2 & -2 & 2 & 0 \\ 0 & -2 & -4 & -2 & -2 \\ 0 & 2 & 1 & -1 & 0 \end{array} \right)$$

$$\Leftrightarrow \begin{matrix} L''_1 \\ L''_2 \\ L''_3 \\ L''_4 \end{matrix} = \begin{matrix} L'_1 \\ L'_2 \\ L'_2 - L'_4 \\ L'_3 + L'_4 \end{matrix} \left(\begin{array}{cccc|c} 2 & 0 & -2 & -2 & 0 \\ 0 & 2 & -2 & 2 & 0 \\ 0 & 0 & -3 & 3 & 0 \\ 0 & 0 & -3 & -3 & -2 \end{array} \right)$$

$$\Leftrightarrow \begin{matrix} L'''_1 \\ L'''_2 \\ L'''_3 \\ L'''_4 \end{matrix} = \begin{matrix} L''_1 \\ L''_2 \\ L''_3 \\ L''_3 - L''_4 \end{matrix} \left(\begin{array}{cccc|c} 2 & 0 & -2 & -2 & 0 \\ 0 & 2 & -2 & 2 & 0 \\ 0 & 0 & -3 & 3 & 0 \\ 0 & 0 & 0 & 6 & 2 \end{array} \right)$$

$$\Leftrightarrow \begin{pmatrix} 2 & 0 & -2 & -2 \\ 0 & 2 & -2 & 2 \\ 0 & 0 & -3 & 3 \\ 0 & 0 & 0 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

$$\Leftrightarrow \begin{cases}
2x_1 - 2x_3 - 2x_4 = 0 & (13) \\
2x_2 - 2x_3 + 2x_4 = 0 & (14) \\
-3x_3 + 3x_4 = 0 & (15) \\
6x_4 = 2 & (16)
\end{cases}$$

De (16) : $x_4 = \frac{1}{3}$ (17), portons (17) dans (15) :

$$-3x_3 + 1 = 0 \Leftrightarrow x_3 = \frac{1}{3} \quad (18)$$

Portons (17) et (18) dans (14) : $2x_2 - \frac{2}{3} + \frac{2}{3} = 0 \Leftrightarrow x_2 = 0$ (19)

En portant (17) et (18) dans (13) : $2x_1 - \frac{2}{3} - \frac{2}{3} = 0 \Leftrightarrow 2x_1 = \frac{4}{3} \Leftrightarrow x_1 = \frac{2}{3}$

Portons (17) dans (8) : $\lambda_2 = -\frac{2}{3}$ et de même (18) dans (7) : $\lambda_1 = -\frac{2}{3}$

D'où $x_1 = \frac{2}{3}, x_2 = 0, x_3 = \frac{1}{3}, x_4 = \frac{1}{3}$ et $\lambda_1 = \lambda_2 = -\frac{2}{3}$.

Par suite $x^* = (x_1, x_2, x_3, x_4) = (\frac{2}{3}, 0, \frac{1}{3}, \frac{1}{3})$ et $\lambda^* = (\lambda_1, \lambda_2) = (-\frac{2}{3}, -\frac{2}{3})$.

Montrons enfin que les conditions d'optimalité sont vérifiées.

En effet, d'une part on a :

$$\begin{aligned}\nabla_x L(x^*, \lambda^*) &= \nabla f(x^*) + A^T \lambda^* \\ &= \begin{pmatrix} 2 \times \frac{2}{3} \\ 2 \times 0 \\ 2 \times \frac{1}{3} \\ \frac{2}{3} \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix}^T \begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix} \\ &= \begin{pmatrix} \frac{4}{3} \\ 0 \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{2}{3} \\ -\frac{2}{3} \end{pmatrix} \\ &= \begin{pmatrix} \frac{4}{3} \\ 0 \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix} + \begin{pmatrix} -\frac{2}{3} - \frac{2}{3} \\ -\frac{2}{3} + \frac{2}{3} \\ -\frac{2}{3} + 0 \\ -\frac{2}{3} \end{pmatrix} \\ &= \begin{pmatrix} \frac{4}{3} - \frac{4}{3} \\ 0 + 0 \\ \frac{2}{3} - \frac{2}{3} \\ \frac{2}{3} - \frac{2}{3} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ &= 0.\end{aligned}$$

Et d'autre part ; l'équation (1.55) demeure vérifiée en considérant au moins

$$Y = \begin{pmatrix} -\frac{1}{2}y_3 - \frac{1}{2}y_4 \\ -\frac{1}{2}y_3 + \frac{1}{2}y_4 \\ y_3 \\ y_4 \end{pmatrix}.$$

En vertu de ce qui précède, il en résulte que le problème (1.74) - (1.75) admet un minimum en $x^* = (\frac{2}{3}, 0, \frac{1}{3}, \frac{1}{3})$ avec $\lambda^* = (-\frac{2}{3}, -\frac{2}{3})$ vérifiant les conditions d'optimalité (1.54)-(1.55).

1.4 Convergence et vitesse de convergence d'un algorithme

Étudier la convergence d'un algorithme, c'est étudier la convergence de la suite des itérés générés par l'algorithme. Un algorithme est dit *convergent* si la suite des itérés $(x_k)_{k \in \mathbb{N}}$ converge vers un point x^* , solution du problème : (1.4). C'est-à-dire

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

En plus, la convergence est dite *locale* si elle n'a lieu que pour des points initiaux x_0 dans un voisinage de x^* ; dans le cas contraire elle est dite *globale*, c'est-à-dire les points initiaux ne sont pas forcément dans le voisinage. ([8] : p29)

En pratique, le but d'un algorithme d'optimisation ne sera que de trouver un point critique (c'est-à-dire un point vérifiant la condition d'optimalité du premier ordre (1.36) : $\nabla f(x^*) = 0$), nous introduisons alors la notion de convergence globale d'un algorithme en optimisation :

Définition 1.26. Convergence globale ([14] : p271)

Soit un algorithme itératif qui génère une suite $(x_k)_{k \in \mathbb{N}}$ dans \mathbb{R}^n , afin de résoudre le problème de minimisation sans contrainte $\min_{x \in X} f(x)$ où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction continûment différentiable, l'algorithme est dit globalement convergent si

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

, quelque soit $x_0 \in \mathbb{R}^n$.

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

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

Définition 1.27. ([8] : p31)

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

1. **Linéaire** si l'erreur $e_k = \|x_k - x^*\|$ décroît linéairement. C'est à dire s'il existe $\tau \in]0, 1[$ tel que :

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = \tau. \quad (1.76)$$

2. **Super linéaire** si

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \quad (1.77)$$

3. **D'ordre p** s'il existe $\tau \geq 0$ tel que :

$$\lim_{k \rightarrow +\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} = \tau. \quad (1.78)$$

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

1.5 Existence et unicité d'un point minimum

([17] : pp21-22)

Théorème 1.8. (*Existence*)

Soit $U \subset \mathbb{R}^n$ un ensemble non vide et fermé et $f : U \rightarrow \mathbb{R}$ une fonction continue. On suppose :

1. Soit U est borné.

2. Soit U est non borné et f est une fonction coercive. (au sens de la définition 1.18)

Alors il existe au moins un point de minimum de f sur U . (i.e $\exists u^* \in U$ tel que $f(u) \leq f(u^*, \forall u \in U)$).

Preuve. On distingue deux cas :

Cas 1 : L'ensemble U est borné.

Alors comme U est aussi fermé, U est compact. Comme f est continue, le Théorème de Weierstrass nous assure que f est bornée sur U et elle atteint ses bornes. Donc il existe au moins un point de minimum absolu de f sur U .

Cas 2 : L'ensemble U est non borné.

Soit $a \in U$ et considérons l'ensemble

$$E = \{x \in U, f(x) \geq f(a)\} \text{ Remarque : } a \in E. \quad (1.79)$$

Il est facile de montrer :

1. **E est fermé**

(car $E = f^{-1}([-\infty, f(a)])$ donc E est l'image inverse d'un intervalle fermé par une fonction continue).

2. **E est borné**

(supposons le contraire : alors il existe une suite $x_k \in E$ avec $\|x_k\| \rightarrow +\infty$ pour $k \rightarrow +\infty$. Comme f est coercive, ceci entraîne $f(x_k) \rightarrow +\infty$ ce qui est absurde, car $f(x_k) \geq f(a), \forall k \in \mathbb{N}$.)

On déduit alors que E est un ensemble compact dans \mathbb{R}^n . Du Théorème de Weierstrass, $\exists u^* \in E$ tel que $f(u^*) \geq f(a), \forall u \in E$. Mais d'autre part, on a $f(u^*) < f(a), \forall u \in U - E$. Ceci prouve que u^* est un point de minimum absolu de f sur U , ce qui finit la preuve. \square

Théorème 1.9. (*Unicité*)

Soit $U \subset \mathbb{R}^n$ un ensemble convexe et $f : U \rightarrow \mathbb{R}$ une fonction strictement convexe. Alors il existe au plus un point de minimum de f sur U .

Preuve. On va raisonner par absurde.

Soient $u_1, u_2 \in U$ avec $u_1 \neq u_2$ deux points de minimum de f sur U . Nous avons donc :

$$f(u_1) = f(u_2) \geq f(u), \forall u \in U. \quad (1.80)$$

Comme f est strictement convexe, on a

$$f\left(\frac{1}{2}u_1 + \frac{1}{2}u_2\right) < \frac{1}{2}f(u_1) + \frac{1}{2}f(u_2) = f(u_1) \quad (1.81)$$

(car $f(u_1) = f(u_2)$) et ceci contredit (1.80). \square

Corollaire 1.2. (*Existence et Unicité*)

Soit $\Omega \subset \mathbb{R}^n$ un ouvert, $U \subset \Omega$ un ensemble fermé et convexe et $f : \Omega \rightarrow \mathbb{R}$ une fonction convexe de classe C^1 sur Ω et fortement convexe sur U . Alors il existe un unique point de minimum de f sur U .

Preuve. Existence :

Il y a deux possibilités :

1. Soit U est borné et alors l'existence est immédiate en vertu du Théorème 1.8.
2. Soit U est non borné et alors comme conséquence de la Proposition de la coercivité d'une fonction on en déduit que f est coercive sur U et donc l'existence résulte encore du Théorème 1.8.

Unicité :

Comme f est strictement convexe sur U , l'unicité est une conséquence immédiate du Théorème 1.9. \square

Synthèse partielle

Ce chapitre consistait à donner les notions de base jugées utiles pour la présentation du reste du travail. Nous avons commencé en premier lieu par donner une petite motivation où le problème d'optimisation a été introduit tout en définissant ses termes connexes. En deuxième lieu, il s'agissait de définir les concepts fondamentaux de l'optimisation. En troisième lieu, les conditions d'optimalité jouant un triple rôle dans le développement des algorithmes : elles procurent une analyse théorique du problème, elles inspirent directement des idées des algorithmes et permettent de déterminer un critère d'arrêt pour les algorithmes itératifs. En quatrième lieu, nous avons parlé de la convergence et vitesse de convergence d'un algorithme. En dernier lieu, nous avons présenté les conditions sous lesquelles on trouve l'existence et unicité d'un point minimum.

Chapitre 2

Méthodes numériques d'optimisation sans contrainte

(L'essentiel de ce chapitre a été inspiré des références bibliographiques suivantes : [7] : pp40-58, [14] : pp227-306, [4] : pp43-63 , [5] : pp22-32 et [8] : pp23-37)

Dans ce chapitre nous parlerons de certaines méthodes numériques d'optimisation non linéaire conçues spécifiquement pour minimiser une fonction quelconque en l'absence des contraintes. Les conditions d'optimalité garantissent qu'on ne peut pas faire une descente à partir d'un minimum.

Ainsi, dans la première section de ce chapitre nous allons commencer par présenter la méthode de Newton locale quelles que soient ses insuffisances.

Dans la deuxième, nous présentons des problèmes quadratiques car ils vont souvent apparaître comme sous-problème dans ce travail.

Dans la troisième section, nous allons définir une classe de méthodes spécialement conçues pour des problèmes de minimisation, appelées méthodes de descente et recherche linéaire se référant surtout aux conditions de Wolfe.

Dans la quatrième et dernière section, les méthodes dites à région de confiance qui, constituent une alternative aux méthodes de descente ; et

Enfin dans la cinquième, nous parlerons de méthodes quasi-Newton venant toujours poser une alternative à la méthode de Newton locale.

2.1 Méthode de Newton locale

([14] : pp241-249, [7] : pp53-58 et [8] : pp35-37)

L'idée de la méthode de Newton locale est de résoudre le système d'équations (1.36) à la page 17 qui définit les conditions nécessaires d'optimalité ; voici son algorithme de calcul :

Algorithme 2.1. Méthode de Newton locale

Objectif : Trouver une approximation de la solution du système $\nabla f(x) = 0$

Entrées :

1. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
2. Le hessien de la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.
3. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.

4. La précision demandée $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$.

Sortie : Une approximation de la solution $x^* \in \mathbb{R}^n$.

Initialisation : $k=0$

Itérations :

1. Calculer d_k solution de $\nabla^2 f(x_k)d_k = -\nabla f(x_k)$.
2. $x_{k+1} = x_k + d_k$.
3. $k = k + 1$.

Critère d'arrêt : Si $\|f(x_k)\| \leq \varepsilon$, alors $x^* = x_k$.

Cependant cette méthode pose un problème selon lequel elle a une convergence locale telle que définie dans la section 1.4, page 28. C'est pourquoi elle peut d'abord facilement diverger lorsque le point de départ est trop éloigné de la solution. En suite elle n'est pas définie lorsque la matrice $\nabla^2 f(x_k)$ n'est pas inversible. Enfin, elle n'établit aucune différence entre les points critiques ; elle ne parvient pas à différencier les minima des maxima, tout comme des points stationnaires, ajoutons en plus que cette méthode a aussi une convergence quadratique (le nombre de décimales exactes est multiplié par 2 à chaque itération).

2.2 Problèmes quadratiques

([14] : pp227-240 et [7] : pp40-90)

Ces problèmes consistent à résoudre le problème

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T Qx + b^T x + c. \quad (2.1)$$

Or le paramètre c ne pose pas problème sur la valeur optimale de f , c'est pourquoi le problème (2.1) sera ramené à

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T Qx + b^T x \quad (2.2)$$

où Q est une matrice symétrique $n \times n$ définie positive, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. Nous avons toujours $\nabla f(x) = Qx + b$ et $\nabla^2 f(x) = Q$.

En vertu du théorème 1.5 le problème n'a de solution que si Q est définie positive. L'unique minimum global de (2.2) peut être obtenu en résolvant le système d'équations linéaires

$$Qx = -b \quad (2.3)$$

2.2.1 Résolution directe

Ici nous présentons un algorithme de résolution directe permettant de trouver une solution à (2.2)

Algorithme 2.2. Problèmes quadratiques : Résolution directe

Objectif : Trouver le minimum global de (2.2).

Entrées :

- La matrice $Q \in \mathbb{R}^{n \times n}$ symétrique définie positive .
- Le vecteur $b \in \mathbb{R}^n$.

Sorties : La solution $x^* \in \mathbb{R}^n$.

Résolution

1. Calculer la factorisation de Cholesky $Q = LL^T$
2. Calculer y^* solution du système triangulaire inférieur $Ly = -b$.
3. Calculer x^* solution du système triangulaire supérieur $L^T x = y^*$.

2.2.2 Méthode des gradients conjugués

La méthode des gradients conjugués est une méthode itérative pour résoudre (2.2). Elle a été découverte indépendamment par Stiefel(1952) et Hestenes (1951), qui l'ont finalisée et publiée ensemble.

En effet, nous allons décrire cette méthode en deux temps; d'une part en présentant d'une manière générale la méthode de directions conjuguées et d'autre part la méthode de gradients conjugués proprement dite.

Définition 2.1. (Directions conjuguées ([5] : p23))

Soit une matrice $Q \in \mathbb{R}^{n \times n}$, définie positive. Les vecteurs non nuls de \mathbb{R}^n d_1, d_2, \dots, d_k sont dits Q -conjugués si $d_i^T Q d_j = 0, \forall i, j$ tels que $i \neq j$. Si $Q = I$, les directions conjuguées sont orthogonales.

Théorème 2.1. Indépendances des directions conjuguées

Soit d_1, \dots, d_k un ensemble des directions non nulles et Q -conjuguées. alors les vecteurs d_1, \dots, d_k sont linéairement indépendants.

Preuve . Supposons par l'absurde qu'il existe $\lambda_1, \dots, \lambda_{k-1}$, non tous nuls, tels que $d_k = \lambda_1 d_1 + \dots + \lambda_{k-1} d_{k-1}$. Dès lors $d_k^T Q d_k = \lambda_1 d_k^T Q d_1 + \dots + \lambda_{k-1} d_k^T Q d_{k-1} = 0$ Car les directions sont Q -conjuguées. Ceci est impossible car d_k est non nulle, et Q est définie positive. ([14] : p229) \square

Théorème 2.2. Méthode de directions conjuguées ([14] : p231)

Soient $d_1, \dots, d_l, l \leq n$, un ensemble de directions Q -conjuguées. Soit $x_1 \in \mathbb{R}^n$ et $M_l = x_1 + \langle d_1, \dots, d_l \rangle = \{x/x = x_1 + \sum_{k=1}^l \lambda_k d_k, \lambda \in \mathbb{R}^l\}$ le sous-espace affine engendré par les directions d_1, \dots, d_l . Alors le minimum global du problème

$$\min_{x \in M_l} = \frac{1}{2} x^T Q x + b^T x \quad (2.4)$$

est

$$x_{l+1} = x_1 + \sum_{k=1}^l \alpha_k d_k \quad (2.5)$$

avec

$$\alpha_k = \operatorname{argmin}_{\alpha} f(x_k + \alpha d_k) = -\frac{d_k^T (Q x_k + b)}{d_k^T Q d_k} \quad (2.6)$$

où α_k est le pas.

Preuve . Considérons la fonction

$$g : \mathbb{R}^l \rightarrow \mathbb{R} : \lambda \rightarrow g(\lambda) = f\left(x_1 + \sum_{i=1}^l \lambda_i d_i\right) \quad (2.7)$$

qui permet de transformer le problème (2.4) en problème sans contrainte $\min_{\lambda \in \mathbb{R}^l} g(\lambda)$ tel que $\frac{\partial g}{\partial \lambda_i}(\lambda) = d_i^T \nabla f(x_1 + \sum_{j=1}^l \lambda_j d_j)$. Par le lemme des itérés y relatif on a :

$$\frac{\partial g}{\partial \lambda_i}(\alpha_1, \dots, \alpha_k) = d_i^T \nabla f(x_l + 1) = 0, \forall i.$$

ainsi $\nabla g(\alpha_1, \dots, \alpha_k) = 0$. De plus, $\frac{\partial g}{\partial \lambda_i}(\alpha_1, \dots, \alpha_k) = d_i^T Q_j$. La matrice des dérivées secondes de g est une matrice diagonale dont les valeurs propres sont strictement positives. Il reste à utiliser les conditions suffisantes d'optimalité (théorème (1.3) et théorème (1.4)) pour montrer que $\alpha_1, \dots, \alpha_l$ est un minimum global de g , et que x_{l+1} défini par (2.5) est un minimum global de (2.4). Le cas particulier où $l = n$ est particulièrement important. \square

Corollaire 2.1. Convergence de la méthode des directions conjuguées

Soit $Q \in \mathbb{R}^{n \times n}$ définie positive. Soient d_1, \dots, d_n , un ensemble de directions Q -conjuguées. Soit $x_1 \in \mathbb{R}^n$ arbitraire. L'algorithme basé sur la récurrence $x_{k+1} = x_k + \alpha_k d_k$ avec $\alpha_k = -\frac{d_k^T(Qx_k + b)}{d_k^T Q d_k}$ identifie le minimum global du problème $\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T Q x + b^T x$ en au plus n itérations.

Preuve. En appliquant le théorème (2.2) avec $l=n$ on montre que x_{l+1} est le minimum global. Par le théorème (2.2), on a $M_l = M_n = \mathbb{R}^n$. \square

Voici alors la méthode proprement dite :

Algorithme 2.3. Méthode de gradients conjugués ([14] : p236)

Objectif : Trouver le minimum global de (2.2) c.à.d

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + b^T x$$

Entrées :

1. Une première approximation x_1 de la solution.
2. La matrice $Q \in \mathbb{R}^{n \times n}$ symétrique définie positive.
3. Le vecteur $b \in \mathbb{R}^n$.

Sortie : La solution $x^* \in \mathbb{R}^n$.

Initialisation : $k = 1, d_1 = -Qx_1 - b$

Itérations :

1. Calculer le pas $\alpha_k = -\frac{d_k^T(Qx_k + b)}{d_k^T Q d_k}$.
2. Calculer l'itéré suivant $x_{k+1} = x_k + \alpha_k d_k$.
3. Calculer $\beta_{k+1} = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)} = \frac{(Qx_{k+1} + b)^T (Qx_{k+1} + b)}{(Qx_k + b)^T (Qx_k + b)}$.
4. Calculer la nouvelle direction $d_{k+1} = -Qx_{k+1} - b + \beta_{k+1} d_k$.
5. $k = k + 1$.

Critère d'arrêt : si $\|\nabla f(x_k)\| = 0$ ou $k = k + 1$, alors $x^* = x_k$.

Une caractéristique très importante de l'algorithme des gradients conjugués est que la matrice Q définissant le problème ne doit jamais être manipulée. Il n'est même pas nécessaire de la stocker. Elle est uniquement utilisée pour calculer des produits matrice-vecteur Qx_k ou Qd_k .

2.3 Méthodes de descente et recherche linéaire

2.3.1 Principe général des méthodes de descente

Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_{k \in \mathbb{N}}$ définie par $x_{k+1} = x_k + \alpha_k d_k$ et telle que $\forall k \in \mathbb{N}, f(x_{k+1}) \leq f(x_k)$. Un tel algorithme est ainsi déterminé par deux éléments : le choix de la direction d_k appelée direction de descente et le choix de la taille du pas α_k à faire dans la direction d_k : cette direction est appelée la recherche linéaire.

Les méthodes de descente consistent en une série d'itérations constituées essentiellement de trois étapes :

1. Trouver une direction d_k telle que $\nabla f(x_k)^T d_k < 0$.
2. Trouver un pas α_k tel que $f(x_k + \alpha_k d_k) < f(x_k)$.
3. Calculer $x_{k+1} = x_k + \alpha_k d_k$.

Parmi ces méthodes nous pouvons énumérer la méthode de la plus forte pente préconditionnée, qui, hélas n'est pas généralement bonne à appliquer dans la mesure où les pas ne doivent pas être choisis seulement aveuglement ; d'où il faut recourir aux conditions de Wolfe. ([4] : p43)

2.3.2 Recherche linéaire

([5] : pp22-31 , [6] : pp24-25, [4] : pp46-60 et [14] : pp251-275)

Le choix du pas répond généralement à deux objectifs souvent contradictoires ; trouver le meilleur pas possible et effectuer le moins de calculs possibles. Ces deux objectifs ont donné naissance à deux grandes familles : les algorithmes à pas fixe ($\alpha_k = \alpha_{k-1}$) et ceux à pas optimal (α_k solution du problème $\min_{\alpha > 0} f(x_k + \alpha d_k)$). Ainsi, un pas de descente acceptable est celui qui fait suffisamment décroître la fonction objectif. D'où l'existence des conditions suivantes appelées conditions de Wolfe qui portent parfois le nom de conditions d'Armijo-Goldstein.

Définition 2.2. Diminution suffisante : première condition de Wolfe ([4] : p47)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable, un point $x_k \in \mathbb{R}^n$, une direction (de descente) $d_k \in \mathbb{R}^n$ telle que $\nabla f(x_k)^T d_k < 0$ et un pas $\alpha_k \in \mathbb{R}, \alpha_k > 0$. On dira que la fonction f diminue suffisamment en $x_k + \alpha_k d_k$ par rapport à x_k si

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k \beta_1 \nabla f(x_k)^T d_k, \text{ avec } 0 < \beta_1 < 1 \quad (2.8)$$

La condition (2.8) s'appelle la première condition de Wolfe d'après Wolfe (1969), ou aussi condition d'Armijo d'après Armijo (1966).

Cette condition (2.8) nous permettra de rejeter des pas, qui, étant trop grands ne fournissent pas une diminution suffisante de la fonction.

Définition 2.3. Progrès suffisant : Deuxième condition de Wolfe ([14] : p261)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable, un point $x_k \in \mathbb{R}^n$, une direction (de descente) $d_k \in \mathbb{R}^n$ telle que $\nabla f(x_k)^T d_k < 0$ et un pas $\alpha_k \in \mathbb{R}, \alpha_k > 0$. On dira que le point $x_k + \alpha_k d_k$ apporte un progrès suffisant par rapport à x_k si

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \beta_2 \nabla f(x_k)^T d_k, \text{ avec } 0 < \beta_2 < 1. \quad (2.9)$$

La condition (2.9) s'appelle la seconde condition de Wolfe, d'après Wolfe (1969).

Ainsi, la recherche linéaire permet d'identifier un pas qui vérifie les conditions de Wolfe (2.8) et (2.9). Voici alors l'algorithme de la recherche linéaire :

Algorithme 2.4. Recherche linéaire ([14] : p263)

Objectif : Trouver un pas α^* vérifiant les conditions de wolfe (2.8) et (2.9).

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Un vecteur $x \in \mathbb{R}^n$.
4. Une direction de descente d telle que $\nabla f(x)^T d < 0$.
5. Une première approximation de la solution $\alpha_0 > 0$.
6. Des paramètres β_1 et β_2 tels que $0 < \beta_1 < \beta_2 < 1$ (par exemple $\beta_1 = 10^{-4}$ et $\beta_2 = 0,99$).
7. un paramètre $\lambda > 1$.

Sortie : Un pas α^* tel que les conditions (2.8) et (2.9) soient vérifiées.

Initialisation : $i = 0, \alpha_l = 0, \alpha_r = +\infty$.

Itérations :

1. Si α_i vérifie les conditions (2.8) et (2.9) alors $\alpha^* = \alpha_i$. STOP
2. Si α_i viole (2.8), alors le pas est trop et $\alpha_r = \alpha_i$, $\alpha_{i+1} = \frac{\alpha_l + \alpha_r}{2}$.
3. Si α_i ne viole pas (2.8) et viole (2.9), alors le pas est trop court et $\alpha_l = \alpha_i$,

$$\alpha_{i+1} = \begin{cases} \frac{\alpha_l + \alpha_r}{2} & \text{si } \alpha_r < +\infty \\ \lambda \alpha_i & \text{si non} \end{cases}.$$
4. $i = i + 1$.

2.3.3 Méthode de la plus forte pente

Cette méthode est une version complète et opérationnelle combinant la méthode de la plus forte pente préconditionnée et la recherche linéaire.

Algorithme 2.5. Méthode de la plus forte pente ([14] : p266)

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

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.
4. La précision demandée $\varepsilon \in \mathbb{R}, \varepsilon > 0$.

Sortie : Une approximation de la solution $x^* \in \mathbb{R}$.

Initialisation : $k = 0$.

Itérations :

1. $d_k = -\nabla f(x_k)$.
2. Déterminer α_k en appliquant la recherche linéaire avec $\alpha_0 = 1$.
3. $x_{k+1} = x_k + \alpha_k d_k$.
4. $k = k + 1$.

Critère d'arrêt : Si $\|\nabla f(x_k)\| \leq \varepsilon$, alors $x^* = x_k$.

2.3.4 Méthode de Newton avec recherche linéaire

La méthode de Newton locale ne fonctionne que lorsque le point de départ est suffisamment proche de la solution, elle a une vitesse de convergence quadratique, plus la fonction est non linéaire et mal conditionnée, plus le point de départ doit être proche de la solution. En plus elle n'établit pas de distinction entre minima, maxima et points stationnaires. Une itération de la méthode de Newton locale est

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k). \quad (2.10)$$

Une itération de la méthode de la plus forte pente préconditionnée est

$$x_{k+1} = x_k - \alpha_k D_k \nabla f(x_k) \quad (2.11)$$

où D_k est définie positive. Dès lors, si $\nabla^2 f(x_k)$ est définie positive et $\alpha_k = 1$ est acceptable (au sens des conditions de Wolfe), une itération de la méthode de Newton locale est exactement une itération de la méthode de la plus forte pente préconditionnée, avec $D_k = \nabla^2 f(x_k)^{-1}$. Par contre si $\nabla^2 f(x_k)$ n'est pas définie positive, on pose $D_k = (\nabla^2 f(x_k) + E)^{-1}$ ceci n'est possible que si E est un multiple de l'identité. Voici alors une méthode simple pour obtenir E ainsi qu'une factorisation de Cholesky de $\nabla^2 f(x_k + E)$. ([4] : pp58-60)

Définition 2.4. Décomposition de Cholesky ([14] : p482)

Soit $A \in \mathbb{R}^{n \times n}$ une matrice symétrique définie positive. La décomposition de Cholesky de A est $A = LL^T$ où $L \in \mathbb{R}^{n \times n}$ est une matrice triangulaire inférieure.

Algorithme 2.6. Factorisation de Cholesky modifiée

Objectif : Modifier une matrice afin de la rendre définie positive.

Entrée : une matrice symétrique $A \in \mathbb{R}^{n \times n}$.

Sortie : Une matrice triangulaire inférieure L et $\tau \geq 0$ tels que $A + \tau L = LL^T$ soit positive.

Initialisation : 1. $k=0$ 2. Si $\min_{ii} a_{ii} > 0$, alors $\tau_k = 0$, sinon, $\tau_k = \frac{1}{2} \|A\|_F$.

Itérations :

1. Calculer la factorisation de Cholesky LL^T de $A + \tau I$.
2. Si la factorisation réussit, STOP.
3. Si non, $\tau_{k+1} = \max(2\tau_k, \frac{1}{2} \|A\|_F)$.
4. $k = k + 1$

Néanmoins, cet algorithme est simpliste et coûteux; il peut nécessiter plusieurs factorisations de Cholesky avant de trouver une matrice définie positive. Voici alors une version complète et opérationnelle de la méthode de la plus forte pente preconditionnée combinant la méthode de Newton locale et la recherche linéaire.

Algorithme 2.7. Méthode de Newton avec recherche linéaire ([4] : p58)

Objectif : Trouver (une approximation d') un minimum local du problème $\min_{x \in \mathbb{R}^n} f(x)$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Le hessien de la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.
4. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.
5. La précision demandée $\varepsilon \in \mathbb{R}, \varepsilon > 0$.

Sortie : une approximation de la solution $x^* \in \mathbb{R}$.

Initialisation : $k = 0$

Itérations :

1. Calculer une matrice triangulaire inférieure L_k et τ tels que $L_k L_k^T = \nabla^2 f(x_k) + \tau I$ en utilisant l'algorithme (2.6).
2. Trouver Z_k en résolvant le système triangulaire $L_k Z_k = \nabla f(x_k)$.
3. Trouver d_k en résolvant le système triangulaire $L_k^T d_k = -Z_k$.
4. Déterminer α_k en appliquant la recherche linéaire avec $\alpha_0 = 1$.
5. $x_{k+1} = x_k + \alpha_k d_k$.
6. $k = k + 1$.

Critère d'arrêt : Si $\| \nabla f(x_k) \| \leq \varepsilon$, alors $x^* = x_k$.

Ainsi, cet algorithme est opérationnel en ce sens que toutes les étapes sont bien définies ; l'apport de cet algorithme est d'obtenir un algorithme qui converge quelque soit le point de départ fourni par l'utilisateur, un tel algorithme est dit globalement convergent (au sens de la définition 1.26).

Néanmoins, un algorithme globalement convergent n'implique pas toujours un minimum global, il peut converger aussi vers un minimum local.

2.4 Méthodes de région de confiance

([5] :pp28-31 et [14] :pp281-292)

Dans cette section nous allons parler des méthodes dites de région de confiance qui, comme les méthodes de descente et recherche linéaire permettent de résoudre les désavantages de la méthode de Newton locale . Les méthodes de région de confiance visent le même objectif en utilisant une approche différente et ont comme objectif de remplacer le problème d'optimisation initial par une suite de sous-problèmes d'optimisation plus simples à résoudre. Dans chaque sous-problème, la fonction objectif f est remplacée par une fonction modèle m (modèle quadratique) à un itéré x_k

$$m_{x_k}(x_k + d_k) = f(x_k) + d_k^T \nabla f(x_k) + \frac{1}{2} d_k^T \nabla^2 f(x_k) d_k \text{ avec } d_k = x - x_k \quad (2.12)$$

Ainsi, la **région de confiance** est définie comme la région à l'intérieur de laquelle la confiance est donnée au modèle quadratique quant à sa qualité à fournir une bonne approximation de la fonction objectif.

Généralement, elle sera définie par son rayon Δ_k et un point x appartiendra à cette région si

$$\| x_k - x \| \leq \Delta_k \quad (2.13)$$

Définition 2.5. Modèle quadratique d'une fonction ([14] : p282)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable. Le modèle quadratique de f en \hat{x} est une fonction $m_{\hat{x}} : \mathbb{R}^n \rightarrow \mathbb{R}$ définie par

$$m_{\hat{x}}(x) = f(\hat{x}) + (x - \hat{x})^T \nabla f(\hat{x}) + \frac{1}{2} (x - \hat{x})^T \nabla^2 f(\hat{x}) (x - \hat{x}). \quad (2.14)$$

En posant $d = x - \hat{x}$, on a :

$$m_{\hat{x}}(\hat{x} + d) = f(\hat{x}) + d^T \nabla f(\hat{x}) + \frac{1}{2} d^T \nabla^2 f(\hat{x}) d. \quad (2.15)$$

Par référence à la fonction quadratique ; $Q = \nabla^2 f(\hat{x})$, $g = \nabla f(\hat{x})$ et $c = f(\hat{x})$

Définition 2.6. Sous-problème de région de confiance ([5] : p31)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable, $\hat{x} \in \mathbb{R}^n$, $m_{\hat{x}}$ le modèle quadratique de f en \hat{x} (définition 2.5) et $\Delta_k > 0$. Le sous-problème de région de confiance est le problème de minimisation suivant :

$$\min_d m_{\hat{x}}(\hat{x} + d) = f(\hat{x}) + d^T \nabla f(\hat{x}) + \frac{1}{2} d^T \nabla^2 f(\hat{x}) d \quad (2.16)$$

sous contrainte

$$\|d\| \leq \Delta_k \quad (2.17)$$

En analysant les conditions d'optimalité, (2.17) s'écrit $\frac{1}{2}(\|d\|^2 - \Delta_k^2) \leq 0$ et le lagrangien du problème est $L(d, \mu) = f(\hat{x}) + d^T \nabla f(\hat{x}) + \frac{1}{2} d^T \nabla^2 f(\hat{x}) d + \frac{\mu}{2}(\|d\|^2 - \Delta_k^2)$. Si d^* est solution du problème de région de confiance, les conditions d'optimalité garantissent qu'il existe $\mu \in \mathbb{R}$ tel que :

$$\nabla_d L(d^*, \mu^*) = \nabla f(\hat{x}) + \nabla^2 f(\hat{x}) d^* + \mu^* d^* = 0, \quad (2.18)$$

$$\mu^* \geq 0 \quad (2.19)$$

et

$$\mu^*(\|d\|^2 - \Delta_k^2) = 0. \quad (2.20)$$

Si la solution d^* se situe strictement à l'intérieur de la région de confiance i.e $\|d^*\| < \Delta_k$ (2.20) assure que $\mu^* = 0$, (2.18) se simplifie et correspond aux conditions nécessaires d'optimalité du problème sans contrainte. Voici d'autres définitions utiles dans cette partie du travail :

Définition 2.7. Point de Newton ([14] : p248)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable et soit $x_k \in \mathbb{R}^n$ tel que $\nabla^2 f(x_k)$ est définie positive. Le point de Newton de f en x_k est le point

$$X_N = x_k + d_N \quad (2.21)$$

où d_N est la solution du système d'équations

$$\nabla^2 f(x_k) d_N = -\nabla f(x_k). \quad (2.22)$$

Le point de Newton minimise le modèle quadratique de la fonction en x_k . Si $\nabla^2 f(x_k)$ est définie positive, il s'agit du minimum du modèle quadratique en x_k . D'autre part, le point minimisant le modèle quadratique dans la direction de la plus forte descente s'appelle le point de Cauchy.

Définition 2.8. Point de Cauchy ([14] : 249)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois différentiable et soit $x_k \in \mathbb{R}^n$. Le point de Cauchy de f en x_k est le point x_C qui minimise le modèle quadratique de f dans la direction de la plus forte descente. C'est-à-dire

$$x_c = x_k - \alpha_c \nabla f(x_k) \quad (2.23)$$

où

$$\alpha_c = \arg \min_{\alpha \in \mathbf{R}_o^+} m_{x_k}(x_k - \alpha \nabla f(x_k)). \quad (2.24)$$

Si f est convexe dans la direction du gradient, en utilisant (2.15) on obtient :

$$\alpha_c = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)} \quad (2.25)$$

2.4.1 Résolution du sous problème de région de confiance

Le problème (2.16)-(2.17) sera résolu approximativement, il existe nombreuses méthodes pour résoudre ce problème, néanmoins nous n'en présentons que deux dans ce présent travail. D'une part, la méthode de la patte de chien ou *dogleg* en anglais ; celle-ci est valable lorsque la matrice hessienne au point courant $\nabla^2 f(\hat{x})$ est définie positive. Et d'autre part la méthode de Steihaug-Toint basée sur la méthode des gradients-conjugués.

2.4.2 Méthode dogleg

L'idée principale de la méthode dogleg est la suivante :

1. Si la région de confiance est très petite , l'approximation de Taylor du premier ordre de la fonction est probablement déjà très bonne, et le terme quadratique joue un rôle mineur. Il est dès lors judicieux de suivre la direction de plus forte descente en direction du point de Cauchy. ([5] : p33)
2. Si la région de confiance est plus large, le terme de second ordre devient significatif, et le point de Newton devient la cible à préférer.
3. Afin de combiner les deux, la méthode dogleg consiste à suivre un chemin qui se dirige d'abord vers le point de Cauchy, pour en suite rejoindre la direction de Newton. Ce chemin est suivi jusqu'au bord de la région de confiance. Si le point de Newton est atteint sans sortir de la région de confiance, cela signifie qu'une itération de Newton locale peut être effectuée.

En supposant d'abord que $\nabla^2 f(x)$ est définie positive, la trajectoire à partir du point courant \hat{x} est définie par $\hat{x} + p(\alpha)$ avec

$$p(\alpha) = \begin{cases} \alpha d_c, & 0 \leq \alpha \leq 1 \\ d_c + (\alpha - 1)(x_d - x_c), & 1 \leq \alpha \leq 2 \\ (\eta(3 - \alpha) + \alpha - 2) d_N, & 2 \leq \alpha \leq 3 \end{cases} \quad (2.26)$$

Nous avons alors intérêt de trouver l'intersection de la trajectoire avec le rayon de la région de confiance.

Algorithme 2.8. Intersection avec la région de confiance ([14] : p288)

Objectif : Trouver l'intersection entre une direction et le bord de la région de confiance.

Entrées :

1. $x \in \mathbb{R}^n$ tel que $\|x\| \leq \Delta$,
2. $d \in \mathbb{R}^n$ tel que $d \neq 0$,
3. $\Delta \in \mathbb{R}$ tel que $\Delta > 0$.

Sortie : Le pas λ tel que $\|x + \lambda d\| = \Delta$.

Coefficients : Calculer $a = d^T d, b = 2x^T d$ et $c = x^T x - \Delta^2$

Racine : $\lambda = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$.

En suite lorsque la matrice $\nabla^2 f(\hat{x})$ n'est pas définie positive, c'est simple. Si la fonction est concave dans la direction de la plus forte pente, C.à.d $\nabla f(\hat{x})^T \nabla^2 f(\hat{x}) \nabla f(\hat{x}) \leq 0$. Alors cette direction peut être suivie jusqu'au bord de la région de confiance. Si la fonction est concave dans la direction de Newton, alors la direction de Cauchy est sélectionnée. D'où l'existence de l'algorithme suivant :

Algorithme 2.9. Méthode dogleg

Objectif : Trouver une approximation de la solution du sous- problème de la région de confiance

$$\min_{d \in \mathbb{R}^n} d^T \nabla f(\hat{x}) + \frac{1}{2} d^T \nabla^2 f(\hat{x}) d$$

sous contrainte $\|d\| \leq \Delta$.

Entrées :

1. La valeur du gradient de la fonction au point courant : $\nabla f(\hat{x}) \in \mathbb{R}^n \neq 0$,
2. La valeur du hessien de la fonction au point courant : $\nabla^2 f(\hat{x}) \in \mathbb{R}^{n \times n}$,
3. Le rayon de la région de confiance .

sortie : La solution approximative d^*

Point de Cauchy

1. Calculer la courbure dans la direction de plus forte pente $\beta = \nabla f(\hat{x})^T \nabla^2 f(\hat{x}) \nabla f(\hat{x})$.
2. Si $\beta \leq 0$, le modèle n'est pas convexe. STOP avec $d^* = -\frac{\Delta}{\|\nabla f(\hat{x})\|} \nabla f(\hat{x})$.
3. En utilisant (2.25), calculer $d_c = -\frac{\alpha}{\beta} \nabla f(\hat{x})$ avec $\alpha = \nabla f(\hat{x})^T \nabla f(\hat{x})$.
4. Si $\|d_c\| \geq \Delta$, le point de Cauchy est hors de la région de confiance ; STOP avec $d^* = \frac{\Delta}{\|d_c\|} d_c$.

Point de Newton

1. Calculer d_N en résolvant $\nabla^2 f(\hat{x}) d_N = -\nabla f(\hat{x})$
2. Si $d_N^T \nabla^2 f(\hat{x}) d_N \leq 0$, le modèle n'est pas convexe, STOP avec le point de Cauchy $d^* = d_c$.
3. Si $d_N \leq \Delta$, le point de Newton est dans la région de confiance .STOP avec $d^* = d_N$.

Point dogleg

1. Calculer $\eta = 0, 2 + \frac{0,8\alpha^2}{\beta|\nabla f(\hat{x})^T d_N|}$ et $d_N = \eta d_N$.
2. Si $\|d_d\| \leq \Delta$, le point dogleg est à l'intérieur de la région de confiance ;STOP avec $d^* = \frac{\Delta}{\|d_N\|} d_N$.

Entre Cauchy et dogleg

1. Calculer le point d'intersection $d_c + \lambda(d_d - d_c)$ entre le segment reliant le point de Cauchy et le point dogleg, avec la frontière de la région de confiance en utilisant l'algorithme 2.8 avec $x = \hat{x} + d_c$ et $d = d_d - d_c$.
2. STOP avec $d^* = d_c + \lambda(d_d - d_c)$.

2.4.3 Méthode de Steihaug-Toint

La méthode des gradients conjugués, présentée par l'algorithme 2.3 est conçue pour minimiser des problèmes quadratiques strictement convexes. Steihaug (1983) et Toint (1981) ont proposé une adaptation de cette méthode afin de résoudre le sous-problème de région de confiance, l'idée de base en est la suivante : A chaque itération de la méthode des gradients conjugués, on teste d'abord si le modèle quadratique est convexe dans la direction d_k . Si ce n'est pas le cas, on suit cette direction jusqu'au bord de la région de confiance, et on arrête les itérations. De plus, dès qu'un itéré est à l'extérieur de la région de confiance, on suit la dernière direction calculée jusqu'au bord de la région de confiance, et on arrête l'algorithme. ([5] : p33)

Dans tous les autres cas, la méthode est appliquée dans sa version originale, d'où alors l'existence de l'algorithme suivant utilisant $Q = \nabla^2 f(x_k)$ et $b = \nabla f(x_k)$.

Algorithme 2.10. Méthode des gradients conjugués tronqués de Steihaug-Toint

Objectif : Trouver une approximation de la solution du sous-problème de région de confiance

$$\min \frac{1}{2} x^T Q x + x^T b$$

sous contrainte $\|x\| \leq \Delta$.

Entrées :

1. La matrice $Q \in \mathbb{R}^{n \times n}$,
2. Le vecteur $b \in \mathbb{R}^n$,
3. Le rayon de la région de confiance Δ .

Sortie : La solution approximative $x^* \in \mathbb{R}^n$

Initialisation : $k = 1, x_1 = 0, d_1 = -b$.

Itérations :

1. Vérifier la courbure de la fonction dans la direction d_k . Si $d_k^T Q d_k \leq 0$ alors $x^* = x_k + \lambda d_k$ où λ est obtenu par l'algorithme 2.8.
2. Calculer le pas $\alpha_k = -\frac{d_k^T (Qx_k + b)}{d_k^T Q d_k}$
3. Calculer l'itéré suivant $x_{k+1} = x_k + \alpha_k d_k$,
4. Si $\|x_{k+1}\| > \Delta$, alors $x^* = x_k + \lambda d_k$ où λ est obtenu par l'algorithme (2.8),
5. Calculer $\beta_{k+1} = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)} = \frac{(Qx_{k+1} + b)^T (Qx_{k+1} + b)}{(Qx_k + b)^T (Qx_k + b)}$,
6. Calculer la nouvelle direction $d_{k+1} = -Qx_{k+1} - b + \beta_{k+1} d_k$,
7. $k = k + 1$.

Critère d'arrêt : Si $\|\nabla f(x_k)\| = 0$, ou $k = k + 1$, alors $x^* = x_k$.

2.4.4 Calcul du rayon de la région de confiance

Le rayon de la région de confiance est déterminé par essais-erreurs ; à la première itération on se choisit une valeur arbitraire $\Delta = 10$. En suite, la qualité de l'approximation de la solution du sous-problème de région de confiance est évaluée, et le rayon de la région de confiance est adapté en fonction de cette évaluation. En supposant que la solution du sous problème de région de confiance soit d^* . Dans ce cas nous pouvons comparer la réduction du modèle $m_{\hat{x}}(\hat{x}) - m_{\hat{x}}(\hat{x} + d^*)$ avec celle de la fonction $f(\hat{x}) - f(\hat{x} + d^*)$. Si le modèle est fiable, ces deux quantités sont parfois supposées proches, nous considérons alors le rapport

$$\rho = \frac{f(\hat{x}) - f(\hat{x} + d^*)}{m_{\hat{x}}(\hat{x}) - m_{\hat{x}}(\hat{x} + d^*)}. \quad (2.27)$$

Ici nous considérons trois cas possibles :

1. ρ est proche de 1, voire plus grand et le modèle est très bon.
2. ρ est proche de 0, voire plus petit, et le modèle est mauvais.
3. ρ est intermédiaire.

À présent nous allons caractériser ces trois cas ci-haut élucidés, par les constantes $\eta_1 = 0,01$ et $\eta_2 = 0,9$ telles que $0 < \eta_1 \leq \eta_2 < 1$. Par suite on a :

1. $\rho \geq \eta_2$, l'adéquation entre le modèle et la fonction semble très bonne, en ce sens que la réduction prédite par le modèle a été quasiment atteinte, voire dépassée.
2. $\eta_1 \leq \eta_2$, l'adéquation entre le modèle et la fonction n'est pas parfaite. Mais ce modèle a tout de même permis de réduire la valeur de la fonction, elle est qualifiée de bonne.
3. $\rho < \eta_1$, l'adéquation entre le modèle et la fonction est mauvaise, en ce sens que soit la réduction de la fonction est négligeable par rapport à la prédiction effectuée par le modèle, soit que la valeur de la fonction a augmenté.

Plusieurs stratégies de mise à jour de la région de confiance utilisant ρ ont alors été proposées :

- Si l'adéquation est très bonne, le rayon est doublé.
- Si l'adéquation est bonne, le rayon n'est pas modifié.
- Si l'adéquation est mauvaise, le rayon est réduite à $\frac{1}{2} \|d^*\|$.

Par suite, voici ci-dessous la méthode de Newton avec région de confiance

Algorithme 2.11. Méthode de Newton avec région de confiance ([14] : p293)

Objectif : Trouver (une approximation d')un minimum local du problème $\min_{x \in \mathbb{R}^n} f(x)$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Le hessien de la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.
4. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.
5. Le rayon de la première région de confiance Δ_0 (par défaut, $\Delta_0 = 10$).
6. La précision demandée $\varepsilon \in \mathbb{R}, \varepsilon > 0$.
7. Les paramètres $0 < \eta_1 < \eta_2 < 1$ (par défaut $\eta_1 = 0,01$ et $\eta_2 = 0,9$).

Sortie : Une approximation de la solution $x^* \in \mathbb{R}$

Initialisation : $k = 0$

Itérations :

1. Calculer d_k en résolvant (approximativement) le sous-problème de région de confiance (2.16) - (2.17), en utilisant la méthode "dogleg" ou la méthode de Steihaug-Toint.
2. Calculer $\rho = \frac{f(x_k) - f(x_k + d_k)}{m_{x_k}(x_k) - m_{x_k}(x_k + d_k)}$.
3. Si $\rho < \eta_1$, alors c'est l'échec. $x_{k+1} = x_k$, $\Delta_{k+1} = \frac{1}{2} \|d_k\|$.
4. Si $\rho > \eta_1$, alors c'est la réussite :
 - (a) $x_{k+1} = x_k + d_k$
 - (b) Si $\rho > \eta_2$, alors c'est très bon. $\Delta_{k+1} = 2\Delta_k$, si non (bon).
 - (c) $\Delta_{k+1} = \Delta_k$.
5. $k = k + 1$

Critère d'arrêt : Si $\|\nabla f(x_k)\| \leq \varepsilon$, alors $x^* = x_k$.

2.5 Méthodes quasi-Newton

Dans cette section, nous allons décrire les méthodes dites de Quasi-Newton dont l'idée principale est de proposer une alternative à la méthode coûteuse de Newton utilisant la matrice hessienne à chaque itération, or cette matrice pose de problèmes pratiques car d'une part le calcul analytique des dérivées secondes et leur implémentation sont souvent

des exercices fastidieux et sujets à erreurs , d'autre part le calcul de cette matrice à chaque itération de l'algorithme peut consommer beaucoup de temps de calcul voire même être pénalisant pour l'efficacité de l'algorithme .

Ainsi, ces méthodes tentent de maintenir la structure de l'algorithme sans toute fois utiliser la matrice hessienne, par contre elles se concentrent donc sur la construction itérative des matrices H_k approchant la matrice hessienne ou des matrices B_k approchant l'inverse de la matrice hessienne .

Par ailleurs, ces méthodes présentent autant de variantes parmi lesquelles nous pouvons énumérer la méthode DFP (DAVIDSON-FLETCHER-POWELL), la méthode BFGS (BROYDEN-FLETCHER-GOLDFARB-SHANNON) et la mise à jour symétrique de rang 1 (SR1) ;

Dans ce présent travail, notre attention portera surtout sur les deux dernières parmi les trois ci-haut énumérées.

2.5.1 Méthode BFGS

Dans le but d'approcher la matrice $\nabla^2 f(\hat{x})$, cette méthode utilise une matrice H_k définie par

$$H_k = H_{k-1} + \frac{(Y_{k-1} - H_{k-1}d_{k-1})d_{k-1}^T}{d_{k-1}^T d_{k-1}} \quad (2.28)$$

avec

$$d_{k-1} = x_k - x_{k-1} \quad (2.29)$$

et

$$Y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}). \quad (2.30)$$

Le principal problème de cette méthode est que la matrice H_k n'est en général pas symétrique, ni définie positive. Par contre ; si on voudrait qu'elle la soit à partir d'une matrice symétrique définie positive H_{k-1} , on adopte la procédure suivante :

1. Calculer la factorisation de Cholesky de $H_{k+1} = L_{k-1}L_{k-1}^T$.
2. Opérer une mise à jour de L_{k-1} afin d'obtenir une matrice A_k .
3. Poser $H_k = A_k A_k^T$ pour obtenir une matrice symétrique définie positive. ([18] : p112)

Définition 2.9. Mise à jour BFGS ([14] : P304)

Soient une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable, et deux itérés x_{k-1} et x_k , tels que $d_{k-1}^T Y_{k-1} > 0$, avec $d_{k-1} = x_k - x_{k-1}$ et $Y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$. Soit une matrice symétrique définie positive $H_{k-1} \in \mathbb{R}^{n \times n}$. La mise à jour BFGS est définie par

$$H_k = H_{k-1} + \frac{Y_{k-1}Y_{k-1}^T}{Y_{k-1}^T d_{k-1}} - \frac{H_{k-1}d_{k-1}d_{k-1}^T H_{k-1}}{d_{k-1}^T H_{k-1} d_{k-1}}. \quad (2.31)$$

Cette mise à jour est évidemment l'œuvre des mathématiciens Broyden, Fletcher, Goldfarb et Shannon qui l'ont découverte indépendamment à la fin des années 60, il est donc à constater qu'elle doit son nom aux initiales des ces auteurs précités. A cette étape, présentons alors son algorithme :

Algorithme 2.12. Méthode quasi-Newton BFGS ([18] : p116)

Objectif : Trouver (une approximation d') un minimum local du problème $\min_{x \in \mathbb{R}^n} f(x)$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable.

2. Le gradient de la fonction $\nabla f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. une première approximation de la solution $x_0 \in \mathbb{R}^n$.
4. Une première approximation de l'inverse du hessien $H_0^{-1} \in \mathbb{R}^{n \times n}$ symétrique définie positive, par défaut $H_0^{-1} = I$.
5. La précision demandée $\varepsilon \in \mathbb{R}, \varepsilon > 0$.

Sortie : Une approximation de la solution $x^* \in \mathbb{R}$

Initialisation : $k=0$

Itérations :

1. Calculer $d_k = -H_k^{-1} \nabla f(x_k)$.
2. Déterminer α_k en appliquant la recherche linéaire avec $\alpha_0 = 1$.
3. $x_{k+1} = x_k + \alpha_k d_k$.
4. $k = k + 1$
5. Mise à jour de H_k^{-1} :

$$H_k^{-1} = \left(I - \frac{\bar{d}_{k-1} Y_{k-1}^T}{\bar{d}_{k-1}^T Y_{k-1}} \right) H_{k-1}^{-1} \left(I - \frac{\bar{d}_{k-1} Y_{k-1}^T}{\bar{d}_{k-1}^T Y_{k-1}} \right) + \frac{\bar{d}_{k-1} \bar{d}_{k-1}^T}{\bar{d}_{k-1}^T Y_{k-1}} \quad (2.32)$$

avec $\bar{d}_{k-1} = \alpha_{k-1} d_{k-1} = x_k - x_{k-1}$ et $Y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.

Critère d'arrêt : Si $\|\nabla f(x_k)\| \leq \varepsilon$, alors $x^* = x_k$.

2.5.2 Mise à jour symétrique de rang 1

La mise à jour BFGS est une mise symétrique de rang 2, i.e la matrice $H_k - H_{k-1}$ est une matrice de rang 2. Il est possible également de définir une mise à jour symétrique de rang 1, i.e telle que $H_k = H_{k-1} + \beta V V^T$ où $V \in \mathbb{R}^n$ et $\beta = 1$ ou -1

Définition 2.10. Mise à jour symétrique SR1 ([14] : p308)

Soient une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable, et deux itérés x_{k-1} et x_k . Soit une matrice $H_{k-1} \in \mathbb{R}^{n \times n}$. La mise à jour SR1 est définie par :

$$H_k = H_{k-1} + \frac{(Y_{k-1} - H_{k-1} d_{k-1})(Y_{k-1} - H_{k-1} d_{k-1})^T}{d_{k-1}^T (Y_{k-1} - H_{k-1} d_{k-1})} \quad (2.33)$$

avec $d_{k-1} = x_k - x_{k-1}$ et $Y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.

Il s'agit d'une mise à jour symétrique de rang 1 (SR1) en anglais. Cette mise à jour n'est définie que si $d_{k-1}^T (Y_{k-1} - H_{k-1} d_{k-1}) \neq 0$. De plus, elle ne génère pas nécessairement une matrice définie positive, et ce même si H_{k-1} l'est. Pour cette raison, il est préférable d'utiliser BFGS dans le cadre des algorithmes basés sur la recherche linéaire. Par contre, dans le cadre des méthodes de région de confiance, la mise à jour SR1 s'avère performante. Ci-après, présentons alors son algorithme.

Algorithme 2.13. Méthode quasi-Newton SR1

Objectif : Trouver (une approximation d') un minimum local du problème $\min_{x \in \mathbb{R}^n} f(x)$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.

4. Une première approximation du hessien $H_0 \in \mathbb{R}^{n \times n}$ symétrique, par défaut $H_0 = I$.
5. Le rayon de la première région de confiance Δ_0 (par défaut, $\Delta_0 = 10$).
6. La précision demandée $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$.
7. Les paramètres $0 < \eta_1 \leq \eta_2 < 1$ (par défaut $\eta_1 = 0.01$ et $\eta_2 = 0.9$).

Sortie : Une approximation de la solution $x^* \in \mathbb{R}$.

Initialisation : $k = 0$

Itérations :

1. Calculer d_k en résolvant (approximativement) le sous-problème de région de confiance

$$\min_{d \in \mathbb{R}^n} f(x_k) d^T \nabla f(x_k) + \frac{1}{2} d^T H_k d$$

sous contrainte $\|d\| \leq \Delta_k$ en utilisant la méthode de steihaug-Toint.

2. Calculer $\rho = \frac{f(x_k) - f(x_k + d_k)}{m x_k(x_k) - m x_k(x_k + d_k)}$.
3. Si $\rho < \eta_1$, alors $x_{k+1} = x_k$ et $\Delta_{k+1} = \frac{1}{2} \|d_k\|$.
4. Si $\rho \geq \eta_1$; $x_{k+1} = x_k + d_k$, si $\rho \geq \eta_2$; alors $\Delta_{k+1} = 2\Delta_k$. Si non $\Delta_{k+1} = \Delta_k$.
5. $k = k + 1$.
6. Poser $\bar{d}_{k-1} = x_k - x_{k-1}$ et $Y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.
7. Si $|\bar{d}_{k-1}(Y_{k-1} - H_{k-1}d_{k-1})| \geq 10^{-8} \|\bar{d}_{k-1}\| \|Y_{k-1} - H_{k-1}d_{k-1}\|$ alors

$$H_k = H_{k-1} + \frac{(Y_{k-1} - H_{k-1}\bar{d}_{k-1})(Y_{k-1} - H_{k-1}\bar{d}_{k-1})^T}{\bar{d}_{k-1}^T (Y_{k-1} - H_{k-1}\bar{d}_{k-1})}.$$

Si non $H_k = H_{k-1}$.

Critère d'arrêt : Si $\|\nabla f(x_k)\| \leq \varepsilon$, alors $x^* = x_k$.

Synthèse partielle

Dans ce chapitre, nous avons élucidé des méthodes de résolution des problèmes d'optimisation non linéaire sans contrainte, nous en avons détaillé cinq grandes classes entre autres : la méthode de Newton locale, les problèmes quadratiques, les méthodes de descente et recherche linéaire, les méthodes à région de confiance et enfin les méthodes de Quasi-Newton. Il n'est pas alors question que nous puissions en indiquer la meilleure ou la pire, le choix à opérer pour l'usage d'une des méthodes précitées dépendra d'un utilisateur à l'autre et surtout de la fonction à optimiser.

Cependant signalons qu'en plus de cela il existe aussi des problèmes des moindres carrés qui apparaissent notamment lorsque l'on désire calibrer les paramètres d'un modèle mathématique en utilisant des données (Ex : L'algorithme de Gauss-Newton, le filtre de Kalman basé sur les observations d'un système réel, la régression linéaire lorsque les observations sont sujettes à des erreurs aléatoires, etc). Mais aussi les méthodes de recherche directe qui sont des algorithmes d'optimisation n'utilisant pas les dérivées. (Ex : L'algorithme de Nelder-Mead, la recherche multi-directionnelle de Torczon), qui, malheureusement n'ont pas été jugées du tout commodes dans ce travail.

Ce chapitre étant ainsi clôturé, il va servir de base au suivant qui va porter sur des méthodes d'optimisation avec contraintes dont nous présentons ci-après le nécessaire.

Chapitre 3

Méthodes numériques d'optimisation avec contraintes

(L'essentiel de ce chapitre a été inspiré des références bibliographiques suivantes : [7] : pp135-219, [8] : pp39-50, [8] : pp69-80, [14] : pp385-468 ; [17] : pp40-59 et [12] : pp27-43)

"sans contrainte, la vie serait facile ; quoi que...."

Dans le deuxième chapitre nous avons parlé de méthodes dites sans contraintes, ces dernières paraissent simples et faciles à appliquer comparativement à celles qui vont faire l'objet de ce chapitre.

En effet, dans ce chapitre nous allons présenter des méthodes de résolution des problèmes d'optimisation non linéaire avec contraintes, elles paraissent difficiles que celles sans contraintes que nous avons développées dans le deuxième chapitre ; heureusement il existe des techniques d'optimisation permettant d'une part de transformer un problème d'optimisation avec contraintes en des sous problèmes sans contraintes ; et d'autre part résoudre systématiquement ces sous-problèmes en respectant rigoureusement les exigences de la méthode choisie.

Ainsi, dans la première section de ce chapitre nous allons présenter une version de la méthode de Newton pour les problèmes avec contraintes basée sur des opérateurs de projection sur ces contraintes .

Dans les deuxième et troisième sections nous parlerons respectivement du lagrangien augmenté et de la programmation quadratique séquentielle. Ci-après, nous en présentons l'essentiel.

3.1 Méthode de Newton avec contraintes

([14] : pp385-389, [7] : pp135-171, [8] : pp45-50 et [4] : pp73-75)

Dans cette section, la méthode de Newton pour l'optimisation sans contrainte est adaptée aux problèmes avec contraintes convexes : $\min_x f(x)$ sous contrainte $x \in X \subseteq \mathbb{R}^n$. Où X est un sous-ensemble convexe fermé de \mathbb{R}^n . Étant donné que dans certaines circonstances la méthode de Newton peut être vue comme une version pré-conditionnée de la méthode de la plus forte pente, nous allons alors procéder de manière similaire pour les problèmes avec contraintes : d'une part nous présentons la méthode en utilisant la direction de la plus forte pente et en suite preconditionner celle-ci de manière appropriée .

3.1.1 Méthode du gradient projeté

([7] : pp135-149 et [8] : pp45-50)

L'idée de base de cette méthode est de suivre la direction de plus forte pente, comme dans le cas sans contrainte. Dès que l'on obtient un point non admissible, on projette celui-ci sur l'ensemble X . Notons $[.]^P$ l'opérateur de projection. à chaque itération, nous générons un point admissible y_k à partir de x_k .

$$y_k = [x_k - \alpha_k \nabla f(x_k)]^P \text{ avec } \gamma_k > 0 \quad (3.1)$$

Montrons que la direction $d_k = y_k - x_k$ (si elle est non nulle) est une direction de descente pour toute valeur de γ_k .

Exemple 3.1. Projection sur un convexe ([4] : p73)

Soit X un ensemble convexe fermé non vide de \mathbb{R}^n , et soit $z \in \mathbb{R}^n$. La projection de z sur X , notée $[z]^P$, est obtenue par le problème d'optimisation suivant : $\min_x f(x) = (x - z)^T(x - z)$ sous contrainte $x \in X$. Comme f est convexe, et que $\nabla f(x) = x - z$, une condition nécessaire et suffisante pour que x^* soit la projection de z sur X est

$$(x^* - z)^T(x - x^*) \geq 0, \forall x \in X. \quad (3.2)$$

Il est donc à signaler que si $z \in X$, alors (3.2) implique $x^* = z$.

Lemme 3.1. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable et $x \subseteq \mathbb{R}^n$ un ensemble convexe fermé. Soit $x_k \in X$ et $x(\gamma) = x_k - \gamma \nabla f(x_k)$, avec $\gamma > 0$. Alors si la direction $d(\gamma) = [x(\gamma)]^P - x_k$ est non nulle, il s'agit d'une direction de descente ; et ce quelque soit $\gamma > 0$.

Preuve . Par l'exemple 3.1, nous avons $\left([x(\gamma)]^P - x(\gamma)\right)^T (x - [x(\gamma)]^P) \geq 0, \forall x \in X$. En particulier, comme $x_k \in X$, nous avons :

$$\begin{aligned} \left([x(\gamma)]^P - x(\gamma)\right)^T (x_k - [x(\gamma)]^P) &= - \left([x(\gamma)]^P - x_k + \gamma \nabla f(x_k)\right)^T d(\gamma) \\ &= - (d(\gamma) + \gamma \nabla f(x_k))^T d(\gamma) \\ &= -d(\gamma)^T d(\gamma) - \gamma d(\gamma)^T \nabla f(x_k) \\ &\geq 0 \end{aligned}$$

Ainsi, $d(\gamma)^T \nabla f(x_k) \leq \frac{-d(\gamma)^T d(\gamma)}{\gamma}$. Comme $d(\gamma) \neq 0$ par hypothèse ; nous avons $d(\gamma)^T \nabla f(x_k) < 0$ et $d(\gamma) = [x(\gamma)]^P - x_k$ est une direction de descente. ([14] : p385) \square

La direction $d(\gamma) = [x(\gamma)]^P - x_k$ possède les propriétés suivantes :

1. Si $d(\gamma) = 0$, alors x_k est un point stationnaire car la condition nécessaire d'optimalité est vérifiée.
2. Si $d(\gamma) \neq 0$, alors $d(\gamma)$ est une direction de descente en vertu du lemme 3.1.
3. Comme x_k et $[x(\gamma)]^P$ sont admissibles, la convexité de X assure que $x_k + \alpha d(\gamma) \in X$ pour tout $0 \leq \alpha \leq 1$ en vertu du théorème de direction admissible dans un convexe .

Ainsi, voici la généralisation de l'algorithme de la plus forte pente (algorithme2.5) :

Algorithme 3.1. Méthode du gradient projeté ([8] : p51)

Objectif : Trouver (une approximation de) la solution du problème

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

avec X convexe fermé et non vide .

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. L'opérateur de projection sur $X : [\cdot]^P$.
4. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.
5. Un paramètre $\gamma > 0$ (par exemple $\gamma = 1$).
6. La précision demandée $\varepsilon \in \mathbb{R}, \varepsilon > 0$.

Sortie : une approximation de la solution $x^* \in \mathbb{R}$.

Initialisation : $k = 0$

Itérations :

1. $y_k = [x_k - \gamma \nabla f(x_k)]^P$.
2. $d_k = y_k - x_k$.
3. Déterminer α_k en appliquant la recherche linéaire (algorithme 2.4).
4. $x_{k+1} = x_k + \alpha_k d_k$.
5. $k = k + 1$

Critère d'arrêt : si $\|d_k\| \leq \varepsilon$, alors $x^* = x_k$.

Néanmoins la recherche de y_k dans l'étape 1 de cet algorithme peut être parfois difficile que le problème initial. En effet, y_k est obtenu en résolvant le problème

$$\begin{aligned} y_k &= \arg \min_{x \in X} \frac{1}{2} \|x_k - \gamma \nabla f(x_k) - x\|^2 \\ &= \arg \min_{x \in X} \frac{1}{2} \|x - x_k\|^2 + \gamma \nabla f(x_k)^T (x - x_k). \end{aligned} \quad (3.3)$$

Il s'agit d'un problème d'optimisation sur un convexe, avec une fonction objectif convexe. Dans certains cas, ce problème est simple à résoudre, notamment la projection sur des contraintes de bornes est triviale. Soit l et $u \in \mathbb{R}$ et $X = x/l \leq x \leq u$ Alors

$$([x_k - \gamma \nabla f(x_k)]^P)_i = \begin{cases} l_i & \text{si } (x_k - \gamma \nabla f(x_k))_i \leq l_i \\ (x_k - \gamma \nabla f(x_k))_i & \text{si } l_i \leq (x_k - \gamma \nabla f(x_k))_i \leq u_i \\ u_i & \text{si } (x_k - \gamma \nabla f(x_k))_i \geq u_i \end{cases}$$

Si l'ensemble X défini par des contraintes linéaires d'égalité, on pose $z = x - x_k$ et (3.3) s'écrit

$$\min_{z/Az=b-Ax_k} \frac{1}{2} z^T z + \gamma \nabla f(x_k)^T z$$

la solution de ce problème est donnée par $z^* = A^T(AA^T)^{-1}(b - A(x_k - \gamma \nabla f(x_k)))$ et donc $y_k = x_k + z^*$ càd

$$y_k = (x_k - \gamma \nabla f(x_k)) + A^T(AA^T)^{-1}(b - A(x_k - \gamma \nabla f(x_k))) \quad (3.4)$$

Enfin, en pratique, il est plus judicieux d'effectuer une recherche linéaire le long de l'arc de projection $[x_k - \gamma \nabla f(x_k)]^P$ plutôt que dans la direction d_k .

3.1.2 Gradient projeté préconditionnée

Ici, nous allons maintenant appliquer la méthode du gradient projeté mais après avoir appliqué au préalable un changement de variables .

Définition 3.1. *Changement de variables*

Soit $x_k \in \mathbb{R}^n$. Soit $M \in \mathbb{R}^{n \times n}$ une matrice inversible. Le changement de variables est l'application linéaire définie par M et transformant x en $x' = Mx$.

Soit alors le problème original $\min_{x \in X} f(x)$ et soit une matrice H définie positive dont la factorisation de Cholesky est LL^T . Définissons $x' = L^T x \Leftrightarrow x = L^{-T} x'$. Dans les nouvelles variables, le problème s'écrit $\min_{x' \in X'} g(x') = f(L^{-T} x')$ avec $X' = \{x' / L^{-T} x' \in X\}$. En utilisant (3.3), le pas 1 de l'algorithme s'écrit

$$y'_k = \arg \min_{x' \in X'} \frac{1}{2} \|x' - x'_k\|^2 + \gamma \nabla g(x'_k)^T (x' - x'_k)$$

pour écrire cette expression dans les variables originales on note $\nabla g(x'_k) = L^{-1} \nabla f(L^{-T} x'_k) = L^{-1} \nabla f(x_k)$ pour obtenir

$$y_k = \arg \min_{x \in X} \frac{1}{2} (L^T x - L^T x_k)^T (L^T x - L^T x_k) + \gamma \nabla f(x_k)^T L^{-T} (L^T x - L^T x_k)$$

ou encore

$$y_k = \arg \min_{x \in X} \frac{1}{2} (x - x_k)^T H (x - x_k) + \gamma \nabla f(x_k)^T (x - x_k) \quad (3.5)$$

A nouveau, le calcul de y_k peut s'avérer difficile. Dans le cas où X est défini uniquement par des équations linéaires, il s'agit du problème quadratique dont la solution analytique est donnée par le théorème 1.7.

Ci-dessous, trouvons alors l'algorithme synthétique.

Algorithme 3.2. *Méthode du gradient projeté préconditionnée*

Objectif : Trouver (une approximation de) la solution du problème

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

avec X convexe fermé et non vide.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Une famille de préconditionneurs $(H_k)_k$ telle que H_k est définie positive pour tout k .
4. Une première approximation de la solution $x_0 \in \mathbb{R}^n$.
5. Un paramètre $\gamma > 0$ (par exemple $\gamma = 1$)
6. La précision demandée $\varepsilon \in \mathbb{R}, \varepsilon > 0$.

Sortie : une approximation de la solution $x^* \in \mathbb{R}$.

Initialisation : $k = 0$

Itérations :

1. Calculer

$$y_k = \arg \min_{x \in X} \frac{1}{2} (x - x_k)^T H (x - x_k) + \gamma \nabla f(x_k)^T (x - x_k)$$

.

2. $d_k = y_k - x_k$
3. Déterminer α_k en appliquant la recherche linéaire avec $\alpha_0 = 1$.
4. $x_{k+1} = x_k + \alpha_k d_k$.
5. $k = k + 1$

Critère d'arrêt : si $\|d_k\| \leq \varepsilon$, alors $x^* = x_k$.

Dans le cas particulier où $H_k = \nabla f(x_k)^2 + \tau I$, avec τ choisi pour que H_k soit définie positive, nous obtenons la méthode de Newton contrainte .

3.2 Lagrangien augmenté

([4] : pp78-80, [7] : pp219-228 et [14] : pp431-435)

Une difficulté des problèmes d'optimisation avec contraintes est la présence de deux critères souvent contradictoires :

1. Réduire la fonction objectif et
2. Identifier des points admissibles.

La méthode du lagrangien augmenté est directement inspirée des conditions d'optimalité de Karush-kuhn-Tucker. L'idée de base consiste à transformer un problème avec contraintes en une suite de problèmes sans contrainte, en pénalisant de plus en plus la violation éventuelle des contraintes.

En effet, dans cette section ; nous considérons le problème d'optimisation (1.8) - (1.9). C.à.d

$$\min_{x \in \mathbb{R}^n} f(x) \quad (3.6)$$

sous contrainte

$$h(x) = 0 \quad (3.7)$$

Par ailleurs, il est toujours possible de transformer une contrainte d'inégalité $g_i \leq 0$ avec $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, en une contrainte d'égalité en introduisant des variables d'écart $z_i \in \mathbb{R}$ pour obtenir $g_i(x) + z_i^2 = 0$. Il est également intéressant de noter que l'approche utilisée par les méthodes du lagrangien augmenté s'oppose à celle utilisée dans le cadre des méthodes de points intérieurs. En effet, ces dernières mettent principalement l'accent sur la génération d'itérés admissibles. Dans les premières itérations, cela se fait au détriment de l'optimalité. Au fur et à mesure que la hauteur des barrières diminue, les algorithmes de points intérieurs convergent vers l'optimum.

Dans le cadre des méthodes du lagrangien augmenté, c'est l'inverse ; l'accent est mis principalement sur l'identification de solutions des sous-problèmes, au besoin en violant les contraintes. Par suite, deux types de pénalisation de la violation des contraintes seront à considérer ici :

1. La pénalisation ou relaxation lagrangienne et
2. La pénalisation quadratique.

3.2.1 Pénalisation Lagrangienne

En se référant à la C.N pour contraintes d'égalité si x^* , λ^* vérifient les conditions suffisantes d'optimalité (1.57) et (1.58), alors x^* est un minimum local du problème

$$\min_{x \in \mathbb{R}^n} L_c(x, \lambda^*) \quad (3.8)$$

avec c suffisamment grand. Ainsi, si l'on connaît la valeur de λ^* , le problème d'optimisation avec contrainte (3.6)-(3.7) se ramène au problème d'optimisation sans contraintes (3.8) et les méthodes d'optimisation sans contraintes vues au deuxième chapitre peuvent s'appliquer ; malheureusement, en pratique, la valeur de λ^* est aussi souvent difficile à obtenir que la valeur de x^* , et cette méthode ne peut pas être utilisée directement ; c'est pourquoi la seconde pénalisation joue un rôle important .

3.2.2 Pénalisation quadratique

La pénalisation quadratique revient à rendre c suffisamment grand pour qu'un point non admissible pour le problème de départ ne puisse pas être optimal pour le lagrangien augmenté . La méthode consiste à résoudre une suite des problèmes sans contrainte

$$x_k = \arg \min_{x \in \mathbb{R}^n} Lc_k(x, \lambda) \quad (3.9)$$

avec λ donné ; et c_k une suite de réels telle que

$$\lim_{k \rightarrow \infty} c_k = +\infty.$$

En général, la solution du problème à l'étape k sert de départ pour le calcul de la solution à l'étape $k + 1$. La bonne nouvelle, c'est que la méthode fonctionne quelque soit λ , la mauvaise nouvelle c'est que ce problème de minimisation sans contrainte devient de plus en plus mal conditionné au fur et à mesure que c_k augmente. Cette situation provoque des problèmes importants pour la résolution des problèmes sans contrainte. Dès lors, il est nécessaire de combiner les deux types de pénalisation afin d'obtenir un algorithme efficace .

3.2.3 Double pénalisation

En utilisant les deux types de pénalisation, on parvient à obtenir les avantages de deux approches, il est important de remarquer que la méthode de pénalisation quadratique fonctionne d'autant mieux que la valeur de λ est proche de λ^* , il s'avère alors important de pouvoir obtenir une bonne approximation de λ^* . Par suite l'algorithme que nous allons décrire est basé non seulement sur une suite des paramètres de pénalité $(c_k)_k$ telle que $c_k \rightarrow \infty$, mais aussi sur une suite de vecteurs $(\lambda_k)_k$. Théoriquement, il suffit que la suite $(\lambda_k)_k$ soit bornée pour que la méthode fonctionne .

Théorème 3.1. Méthode du lagrangien augmenté ([14] : pp436-437)

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ deux fonctions continues et $X \subseteq \mathbb{R}^n$ un sous-ensemble fermé de \mathbb{R}^n tel que l'ensemble $\{x \in X / h(x) = 0\}$ soit non vide. Soit la suite $(c_k)_k$ telle que, pour tout k , $c_k \in \mathbb{R}$ et $0 < c_k < c_{k+1}$. De plus,

$$\lim_{k \rightarrow \infty} c_k = +\infty$$

Soit la suite bornée $(\lambda_k)_k$ telle que $\lambda_k \in \mathbb{R}^m$ pour tout k . Soit x_k le minimum global de

$$\min_{x \in X} Lc_k(x, \lambda_k) = f(x) + \lambda_k^T h(x) + \frac{c_k}{2} \| h(x) \|^2 . \quad (3.10)$$

Alors, tout point limite de la suite $(x_k)_k$ est un minimum global du problème

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

sous contrainte $h(x) = 0$ et $x \in X$

Preuve . Soit f^* la valeur optimale du problème avec contraintes et soit k arbitraire.

$$\begin{aligned} f^* &= \min_{h(x)=0, x \in X} f(x) \\ &= \min_{h(x)=0, x \in X} f(x) + \lambda_k^T h(x) + \frac{c_k}{2} \|h(x)\|^2 \\ &= \min_{h(x)=0, x \in X} Lc_k(x, \lambda_k) \end{aligned}$$

comme X est non vide, et f continue, f^* est fini. Par définition de x_k , minimum global de (3.10), nous avons

$$Lc_k(x_k, \lambda_k) \leq Lc_k(x, \lambda_k), \forall x \in X \quad (3.11)$$

donc en prenant le minimum,

$$Lc_k(x_k, \lambda_k) \leq \min_{h(x)=0, x \in X} Lc_k(x, \lambda_k) = f^*$$

, et ce pour tout k . Lorsque k tend vers l'infini, comme f^* est fini, $Lc_k(x_k, \lambda_k)$ l'est aussi . Soit \bar{x} un point limite de la suite $(x_k)_k$, et $\bar{\lambda}$ un point limite de la suite $(\lambda_k)_k$ (il existe car la suite est bornée). Ainsi, en passant à la limite supérieure, et en tenant compte que les fonctions f et h sont continues, nous obtenons

$$\begin{aligned} \limsup_{k \rightarrow \infty} Lc_k(x_k, \lambda_k) &\leq f^* \\ \limsup_{k \rightarrow \infty} f(x_k) + \lambda_k^T h(x_k) + \frac{c_k}{2} \|h(x_k)\|^2 &\leq f^* \\ f(\bar{x}) + \bar{\lambda}^T h(\bar{x}) + \limsup_{k \rightarrow \infty} \frac{c_k}{2} \|h(x_k)\|^2 &\leq f^* \end{aligned}$$

Pour que le terme de gauche reste fini, alors que $(c_k)_k \rightarrow \infty$, il faut que $(h(x_k))_k \rightarrow 0$, et donc $h(\bar{x}) = 0$, l'expression ci-dessus se simplifie en $f(\bar{x}) \leq f^*$. De plus, comme X est fermé, on a aussi $\bar{x} \in X$. Dès lors \bar{x} est bien un minimum global du problème. \square

Notons que, si x^* est un minimum local du problème avec contrainte (3.6) -(3.7) alors il existe $X \subseteq \mathbb{R}^n$, fermé, tel que x^* est un minimum global pour le problème $\min_{x \in X} f(x)$ sous contrainte $h(x) = 0$, et le théorème s'applique. Le résultat ci-après nous permet de trouver une approximation de λ^* :

Théorème 3.2. Approximation des multiplicateurs de Lagrange ([14] : p438)

Soient f et h deux fonctions continûment différentiables, soit la suite $(c_k)_k$ telle que, pour tout k , $c_k \in \mathbb{R}$ et $0 < c_k < c_{k+1}$. De plus,

$$\lim_{k \rightarrow \infty} c_k = +\infty$$

. Soit la suite bornée $(\lambda_k)_k$ telle que $\lambda_k \in \mathbb{R}^m$ pour tout k . Soit la suite $(\varepsilon_k)_k$ telle que $\varepsilon_k > 0$ pour tout k et $\lim_{k \rightarrow \infty} \varepsilon_k = 0$. Soit la suite $(x_k)_k$ telle que

$$\|\nabla_x Lc_k(x_k, \lambda_k)\| \leq \varepsilon_k \quad (3.12)$$

Soit $(x_k)_{k \in K}$ une sous-suite de la suite $(x_k)_k$, convergeant vers x^* . Si $\nabla h(x^*)$ est de rang plein, alors

$$\lim_{k \in K, k \rightarrow \infty} \lambda_k + c_k h(x_k) = \lambda^* \quad (3.13)$$

où x^* et λ^* vérifient les conditions nécessaires d'optimalité du premier ordre. C.à.d

$$\nabla f(x^*) + \nabla h(x^*)\lambda^* = 0 \quad (3.14)$$

et

$$h(x^*) = 0. \quad (3.15)$$

Preuve . Supposons sans perdre de généralité, que la suite $(x_k)_k$ converge vers x^* (en éliminant tous les termes tels que $k \notin K$) notons

$$l_k = \lambda_k + c_k h(x_k). \quad (3.16)$$

En utilisant le lagrangien augmenté on a :

$$\begin{aligned} \nabla_x L(x_k, \lambda_k) &= \nabla f(x_k) + \nabla h(x_k)\lambda_k + c_k \nabla h(x_k)h(x_k) \\ &= \nabla f(x_k) + \nabla h(x_k)(\lambda_k + c_k h(x_k)) \text{ et par (3.16) on obtient} \\ \nabla_x L(x_k, \lambda_k) &= \nabla f(x_k) + \nabla h(x_k)l_k. \end{aligned} \quad (3.17)$$

Par continuité, comme $\nabla h(x^*)$ est de rang plein, à partir de k suffisamment grand, $\nabla h(x_k)$ est de rang plein également. Dès lors $\nabla_x L(x_k, \lambda_k) = \nabla f(x_k) + \nabla h(x_k)l_k$
 $\nabla h(x_k)^T \nabla_x L(x_k, \lambda_k) = \nabla h(x_k)^T \nabla f(x_k) + \nabla h(x_k)^T \nabla h(x_k)l_k$.
Par (3.12), comme $\varepsilon_k \rightarrow 0$, on a $\nabla_x L(x_k, \lambda_k) \rightarrow 0$, lorsque k tend vers l'infini, on obtient

$$\lambda^* = - \left(\nabla h(x^*)^T \nabla h(x^*) \right)^{-1} \nabla h(x^*)^T \nabla f(x^*) \quad (3.18)$$

en faisant tendre k vers l'infini dans (3.17), on obtient (3.14) $\nabla f(x^*) + \nabla h(x^*)\lambda^* = 0$, comme $l_k = \lambda_k + c_k h(x_k) \rightarrow \lambda^*$, et que $(\lambda_k)_k$ est bornée, alors $(c_k h(x_k))_k$ est aussi bornée. comme $c_k \rightarrow \infty$, alors $h(x_k) \rightarrow 0$, et on obtient (3.15) $h(x^*) = 0$ \square

Ce résultat nous permet donc de définir la suite $(\lambda_k)_k$ de la manière suivante :

$$\lambda_{k+1} = \lambda_k + c_k h(x_k). \quad (3.19)$$

Ce théorème sert maintenant de base pour définir l'algorithme. En effet, il suffit de préciser les suites de paramètres de pénalité $(c_k)_k$, de paramètres $(\lambda_k)_k$, et de paramètres ε_k pour appliquer cet algorithme. A chaque itération

1. On résout $\min_x L(x, \lambda_k)$ à une précision ε_k , en utilisant un algorithme approprié du deuxième chapitre afin d'obtenir x_{k+1} .
2. Si x_{k+1} est suffisamment admissible, on met à jour les multiplicateurs de Lagrange en utilisant (3.13).
3. Si non, on augmente le paramètre de pénalité c_k .

A présent ; voici alors l'algorithme proprement dit :

Algorithme 3.3. Algorithme du lagrangien augmenté

Objectif : Trouver un minimum local du problème (1.8)-(1.9). i.e $\min_{X \in \mathbb{R}^n} f(x)$ sous contrainte $h(x) = 0$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Le hessien de la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.

4. La fonction $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ deux fois différentiable.
5. La matrice gradient de la fonction $\nabla h : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$.
6. Pour chaque contrainte $i = 1, \dots, m$ le hessien de la fonction $\nabla^2 h_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$.
7. Une solution initiale $(x_0, \lambda_0)^T$.
8. Un paramètre de pénalité initial c_0 (par défaut $c_0 = 10$)
9. La précision demandée $\varepsilon > 0$.

Sortie : Une approximation de la solution (x^*, λ^*)

Initialisation : $k = 0, \hat{\eta}_0 = 0, 1258925, \tau = 10, \alpha = 0, 1; \beta = 0, 9; \varepsilon_k = 1/c; \eta_k = \hat{\eta}_0/c^\alpha$

Itérations :

1. Utiliser la méthode de Newton avec recherche linéaire ou avec région de confiance pour résoudre

$$x_{k+1} = \arg \min_{X \in \mathbb{R}^n} Lc_k(x, \lambda_k) = f(x) + \lambda_k h(x) + \frac{c_k}{2} \|h(x)\|^2 \quad (3.20)$$

en utilisant x_k comme point de départ; et ε_k comme précision.

2. Si $\|H(x_k)\| \leq \eta_k$, alors on met à jour les multiplicateurs de Lagrange :
 - (a) $\lambda_{k+1} = \lambda_k + c_k h(x_k)$
 - (b) $c_{k+1} = c_k$
 - (c) $\varepsilon_{k+1} = \varepsilon_k / c_k$
 - (d) $\eta_{k+1} = \eta_k / c_k^\beta$
3. Si $\|H(x_k)\| > \eta_k$, alors on met à jour le paramètre de pénalité :
 - (a) $\lambda_{k+1} = \lambda_k$
 - (b) $c_{k+1} = \tau c_k$
 - (c) $\varepsilon_{k+1} = \varepsilon_k / c_{k+1}$
 - (d) $\eta_{k+1} = \hat{\eta}_0 / c_{k+1}^\alpha$
4. $k = k + 1$

Critère d'arrêt : $\|\nabla L(x_k, \lambda_k)\| \leq \varepsilon$ et $\|h(x_k)\|^2 \leq \varepsilon$.

3.3 Programmation quadratique séquentielle

([4] : p75, [14] : pp451-458)

Dans cette section, comme dans la précédente nous considérons toujours le problème d'optimisation (1.8)- (1.9). C'est-à-dire

$$\min_{x \in \mathbb{R}^n} f(x) \quad (3.21)$$

sous contrainte $h(x) = 0$. Pour décrire cette méthode nous y allons en présentant d'abord la méthode locale de Newton et pour chuter avec celle globalement convergente :

3.3.1 Programmation quadratique séquentielle locale

Nous devons arriver à trouver des variables primales x^* et celles duales λ^* telles que le gradient du lagrangien du problème s'annule. C.à.d $\nabla L(x^*, \lambda^*) = 0$ avec $L(x, \lambda) = f(x) + \lambda^T h(x)$. Nous avons

$$\nabla L(x, \lambda) = \begin{pmatrix} \nabla f(x) + \nabla h(x)\lambda \\ h(x) \end{pmatrix} = \begin{pmatrix} \nabla_x L(x, \lambda) \\ h(x) \end{pmatrix} \quad (3.22)$$

$$\nabla^2 L(x, \lambda) = \begin{pmatrix} \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x) & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix} = \begin{pmatrix} \nabla_{xx}^2 L(x, \lambda) & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix}.$$

Ainsi; une itération k de la méthode de Newton consiste à trouver une direction $d \in \mathbb{R}^{n+m}$ telle que $\nabla^2 L(x, \lambda)d = -\nabla L(x, \lambda)$. C.à.d trouver $d_x \in \mathbb{R}^n$ et $d_\lambda \in \mathbb{R}^m$ tels que

$$\begin{cases} \nabla_{xx}^2 L(x, \lambda)d_x + \nabla h(x)d_\lambda = -\nabla_x L(x, \lambda) \\ \nabla h(x)^T d_x = -h(x) \end{cases}. \quad (3.23)$$

Il est intéressant de constater que le système d'équations (3.23) constitue les conditions d'optimalité du problème quadratique suivant :

$$\min_d \nabla_x L(x_k, \lambda_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 L(x_k, \lambda_k) d \quad (3.24)$$

sous contrainte

$$\nabla h(x_k)^T d + h(x_k) = 0 \quad (3.25)$$

dont le lagrangien est $L^{PQ}(d, l) = \nabla_x L(x_k, \lambda_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 L(x_k, \lambda_k) d + l^T (\nabla h(x_k)^T d + h(x_k))$ et les C.N d'optimalité sont

$$\begin{cases} \nabla_d L^{PQ}(d^*, l^*) = \nabla_x L(x_k, \lambda_k)^T + \nabla_{xx}^2 L(x_k, \lambda_k) d^* + \nabla h(x_k) l^* = 0 \\ \nabla_l L^{PQ}(d^*, l^*) = \nabla h(x_k)^T d^* + h(x_k) \end{cases}$$

Dans le cadre de l'optimisation sans contrainte, le calcul du pas de Newton revenait à l'optimisation d'un modèle quadratique de la fonction alors que dans le cas de l'optimisation avec contrainte, le calcul du pas de Newton revient à optimiser une fonction quadratique sous contraintes linéaires. C'est le problème (3.24)-(3.25). En utilisant (3.22), les équations de Newton (3.23) peuvent s'écrire

$$\begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_k + \nabla h(x_k) d_\lambda = -\nabla f(x_k) - \nabla h(x_k) \lambda_k \\ \nabla h(x_k)^T d_k = -h(x_k) \end{cases}.$$

En posant $\hat{d}_\lambda = d_\lambda + \lambda_k$ on obtient

$$\begin{cases} \nabla_{xx}^2 L(x_k, \lambda_k) d_x + \nabla h(x_k) \hat{d}_\lambda = -\nabla f(x_k) \\ \nabla h(x_k)^T d_x = -h(x_k) \end{cases}. \quad (3.26)$$

Il s'agit également des conditions d'optimalité d'un problème quadratique

$$\min_d \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 L(x_k, \lambda_k) d \quad (3.27)$$

sous contrainte

$$\nabla h(x_k)^T d + h(x_k) = 0. \quad (3.28)$$

Par le théorème 1.7 la solution de ce problème quadratique est

$$\lambda^* = H^{-1} \left(h(x_k) - \nabla h(x_k)^T \nabla_{xx}^2 L(x_k, \lambda_k)^{-1} \nabla f(x_k) \right) \quad (3.29)$$

avec $H = \nabla h(x_k)^T \nabla_{xx}^2 L(x_k, \lambda_k)^{-1} \nabla f(x_k)$ et

$$x^* = -\nabla_{xx}^2 L(x_k, \lambda_k)^{-1} (\nabla h(x_k) \lambda^*) + \nabla f(x_k). \quad (3.30)$$

Cette solution est coûteuse, il faut recourir donc à des algorithmes appropriés. La résolution des équations de karush-kuhn-Tucker par méthode de Newton revenant à résoudre une suite de problèmes, ou programmes, quadratiques; l'algorithme ainsi obtenu s'appelle l'algorithme de programmation quadratique séquentielle, on utilise pour la désigner l'acronyme anglais SQP signifiant "sequential Quadratic Programming"; qui hélas n'est pas globalement convergente : d'où l'ajout de l'adjectif "local" pour la décrire.

Algorithme 3.4. Algorithme SQP local

Objectif : Trouver un minimum local du problème (1.8) - (1.9), $\min_{x \in \mathbb{R}^n} f(x)$ sous contrainte $h(x) = 0$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable.
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Le hessien de la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.
4. La fonction $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ différentiable.
5. La matrice gradient de la fonction $\nabla h : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$.
6. Pour chaque contrainte $i = 1, \dots, m$, le hessien de la fonction $\nabla^2 h_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$.
7. Une solution initiale $(x_0, \lambda_0)^T$.
8. La précision demandée $\varepsilon > 0$.

Sortie : Une approximation de la solution (x^*, λ^*) .

Initialisation : $k = 0$

Itérations :

1. Calculer $\nabla_{xx}^2 L(x_k, \lambda_k) = \nabla^2 f(x_k) + \sum_{i=1}^m (\lambda_k)_i \nabla^2 h_i(x_k)$.
2. Obtenir d_x et d_λ en résolvant le problème quadratique $\min_d \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 L(x_k, \lambda_k) d$ sous contrainte $\nabla h(x_k)^T d + h(x_k) = 0$, avec un algorithme approprié. Pour illustrer l'algorithme, on pourra utiliser (3.29) et (3.30)
3. $x_{k+1} = x_k + d_x$
4. $\lambda_{k+1} = d_\lambda$
5. $k = k + 1$

Critère d'arrêt : si $\| \nabla L(x_k, \lambda_k) \| \leq \varepsilon$.

3.3.2 Algorithme globalement convergent

Afin de rendre la méthode SQP globalement convergente, on s'inspire des méthodes de descente. Dans le cadre de l'optimisation sans contraintes ; il s'agissait d'identifier une direction de descente et de calculer un pas approprié dans cette direction, de manière que le nouvel itéré soit meilleur que son précédent et par là donc ; le concept meilleur correspondait à une diminution suffisante de la fonction objectif .

Dans le cadre de l'optimisation avec contraintes, deux objectifs doivent être atteints : diminuer la valeur de la fonction objectif et vérifier les contraintes. Néanmoins ces deux objectifs sont souvent en conflit dans le cas où, pour essayer de vérifier les contraintes, il faut en général augmenter la valeur de la fonction. Heureusement l'algorithme du lagrangien augmenté essaie de palier à ce conflit dans le sens où la fonction et les contraintes étaient réunies dans une seule fonction objectif que l'on cherchait à minimiser .Ainsi, une telle fonction permettant de mesurer le progrès d'un algorithme tant au niveau de la fonction objectif que des contraintes est appelée **fonction de mérite**. Elle sera dite exacte si toute solution du problème d'optimisation avec contrainte (3.21)-(3.22) est un minimum local de la fonction de mérite .

Définition 3.2. *Fonction de mérite exacte ([14] : p459)*

Soit le problème d'optimisation avec contraintes (1.8)-(1.11). Une fonction $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction de mérite exacte du problème si chaque minimum local x^* du problème (1.8)-(1.11) est aussi un minimum local de la fonction ϕ , sans contrainte.

Pour le problème (3.21)-(3.22), la fonction de mérite exacte la plus utilisée est

$$\begin{aligned}\phi_c(x) &= f(x) + c \| h(x) \| \\ &= f(x) + c \sum_{i=1}^m | h_i | \text{ avec } c > \max_{i=1,\dots,m} | \lambda_i^* | .\end{aligned}\quad (3.31)$$

Pour rendre l'algorithme (3.4) globalement convergeant, nous allons utiliser les mêmes idées que pour les cas sans contrainte, où la fonction de mérite jouera le même rôle de la fonction objectif lorsque la notion de "meilleur" est nécessaire. Les méthodes de recherche linéaire basées sur les conditions de Wolfe et les méthodes de région de confiance peuvent être utilisées dans ce contexte. Nous n'avons pas besoin du gradient lui-même mais seulement de la dérivée directionnelle et il est important que cette dernière soit négative afin que d_k soit une direction de descente pour la fonction de mérite .

Définition 3.3. *Dérivée directionnelle pour la fonction de mérite*

Soient d_x et d_λ vérifiant les conditions (3.26), alors la dérivée directionnelle de ϕ_c dans la direction d_x est

$$\phi'_c(x_k, d_x) = \nabla f(x_k)^T d_x - c \| h(x_k) \|_1 \quad (3.32)$$

Définition 3.4. *Direction de descente pour la fonction de mérite*

Soient d_x et d_λ vérifiant les conditions (3.26), alors la dérivée directionnelle de ϕ_c dans la direction d_x , notée $\phi'_c(x_k, d_k)$ est telle que

$$\phi'_c(x_k, d_k) \leq d_x^T \nabla^2 L(x_k, \lambda_k) d_x - (c - \| d_\lambda \|_\infty) (\| h(x_k) \|_1) .$$

Ainsi, d_x est une direction de descente pour la fonction de mérite si

$$c > \| d_\lambda \|_\infty - \frac{d_x^T \nabla_{xx}^2 L(x_k, \lambda_k) d_x}{\| h(x_k) \|_1} . \quad (3.33)$$

Si $\nabla_{xx}^2 L(x_k, \lambda_k)$ est définie positive il suffit que $c > \| d_\lambda \|_\infty$.

Voici alors son algorithme proprement dit :

Algorithme 3.5. Algorithme SQP globalisé

Objectif : Trouver un minimum local du problème (1.8) - (1.9), $\min_{x \in \mathbb{R}^n} f(x)$ sous contrainte $h(x) = 0$.

Entrées :

1. La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois différentiable .
2. Le gradient de la fonction $\nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
3. Le hessien de la fonction $\nabla^2 f : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$
4. La fonction $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ deux fois différentiable .
5. La matrice gradient de la fonction $\nabla h : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$
6. Pour chaque contrainte $i = 1, \dots, m$, le hessien de la fonction $\nabla^2 h_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$
7. Un paramètre $0 < \beta_1 < 1$ (par défaut $\beta_1 = 0.3$).
8. Un paramètre $\bar{c} > 0$ (par défaut : $\bar{c} = 0.1$).
9. Une solution initiale $(x_0, \lambda_0)^T$.
10. La précision demandée $\varepsilon > 0$.

Sortie : Une approximation de la solution (x^*, λ^*) .

Initialisation : $k = 0, c_0 = \|\lambda_0\|_\infty + \bar{c}$

Itérations :

1. Calculer $\nabla_{xx}^2 L(x_k, \lambda_k) = \nabla^2 f(x_k) + \sum_{i=1}^m (\lambda_k)_i \nabla^2 h_i(x_k)$
2. Trouver une approximation définie positive H_k de $\nabla_{xx}^2 L(x_k, \lambda_k)$ en utilisant la factorisation de Cholesky modifiée .
3. Obtenir d_x et d_λ en résolvant le problème quadratique

$$\min_d \nabla f(x_k)^T d + \frac{1}{2} d^T H_k d \quad (3.34)$$

sous contrainte

$$\nabla h(x_k)^T d + h(x_k) = 0 \quad (3.35)$$

avec un algorithme approprié, pour illustrer la méthode on pourra utiliser (3.29) et (3.30)

4. $c^+ = \|d_\lambda\|_\infty + \bar{c}$
5. Mise à jour du paramètre de pénalité :
 - (a) Si $c_k \geq 1.1c^+$, alors $c_{k+1} = \frac{1}{2}(c_k + c^+)$
 - (b) Si $c^+ \leq c_k < 1.1c^+$, alors $c_{k+1} = c_k$
 - (c) Si $c^+ > c_k$, alors $c_{k+1} = \max(1.5c_k, c^+)$
6. Calculer $\phi'_c(x_k, d_x) = \nabla f(x_k)^T d_x - c \|h(x_k)\|_1$
7. Calcul du pas
 - (a) $i = 0, \alpha_i = 1$
 - (b) Tant que $\phi_c(x_k + \alpha_i d_k) > \phi(x_k) + \alpha_i \beta_1 \phi'_c(x_k, d_x)$, poser $\alpha_{i+1} = \alpha_i/2$ et $i = i + 1$
 - (c) $\alpha = \alpha_i$
8. $x_{k+1} = x_k + \alpha d_x$.
9. $\lambda_{k+1} = d_\lambda$.
10. $k = k + 1$

Critère d'arrêt : $\|\nabla L(x_k, \lambda_k)\| \leq \varepsilon$.

Conclusion Partielle

Dans chapitre, nous avons parlé des méthodes d'optimisation avec contraintes ; nous avons détaillé d'abord la méthode de Newton contrainte , ensuite celle du Lagrangien augmenté et enfin la Programmation quadratique séquentielle.

Néanmoins, en plus de ces classes ci-haut détaillées, il existe d'autres qui, malheureusement n'ont pas été décrites dans ce chapitre pour certaines raisons lacunaires constatées chez l'une ou l'autre relativement à la délimitation du sujet, en voici alors quelques unes :

La *méthode de relaxation* : Elle est bien définie au moins dans la cas d'une fonction elliptique.

La *méthode d'UZAWA* ; basée sur la formulation duale et la recherche des points selles, elle est bien adaptée pour des ensembles de contraintes (définies par des inégalités larges).

La *méthode du plan sécant* ; destinée à résoudre les problèmes de programmation convexe a comme idée fondamentale d'approcher le domaine admissible par un polyèdre qui diminue d'un pas d'itération à l'autre et approche toujours mieux le domaine admissible dans le voisinage de la solution.

La *méthode de linéarisation* à son tour a comme trait distinctif de prendre en considération des contraintes non linéaires de type égalité, ce qui constitue une difficulté insurmontable pour la plus part d'autres méthodes.

La *méthode du simplexe* est probablement l'algorithme la plus célèbre en optimisation conçu pour résoudre les problèmes d'optimisation linéaire.

Enfin, la *méthode des points intérieurs* qui vient corriger le défaut de la méthode du simplexe qui, parfois fournit un nombre exponentiellement grand d'itérations, néanmoins ces méthodes des points intérieurs à leur tour sont motivées par le biais des méthodes barrières.

Ceci étant, nous passons ainsi au quatrième et dernier chapitre où nous tenterons d'expérimenter certaines des méthodes développées dans le chapitres 2.

Chapitre 4

Applications numériques de quelques méthodes d'optimisation sans contrainte

En effet, dans le premier chapitre nous avons surtout défini des concepts fondamentaux et avons présenté certains résultats nécessaires à la compréhension du deuxième et troisième chapitres. Par suite dans les chapitres 2 et 3 nous avons détaillé respectivement des méthodes d'optimisation sans contrainte et celles avec contraintes.

La particularité de ce chapitre est d'expérimenter alors certaines de ces méthodes précédemment développées dans le chapitre 2, notamment la méthode de la plus forte pente et la méthode de Newton.

La plupart des méthodes de résolution de problèmes d'optimisation, telles que présentées consistent à essayer de faire converger une suite de points vers un minimum local. Pour pouvoir comparer ces méthodes les unes aux autres, trois caractéristiques principales se distinguent.

1. **La robustesse** : on veut que la méthode fonctionne dans le plus grand nombre de cas possible, pour le plus grand ensemble possible de points initiaux.
2. **La précision** : on veut que la méthode, malgré les erreurs d'approximation et les erreurs d'arrondi accumulées lors de l'exécution de l'algorithme par un ordinateur (qui fait donc les calculs avec une précision donnée), donne un résultat proche du point de minimum recherché.
3. **La vitesse** : on veut que la méthode converge rapidement vers un point de minimum.

Ces trois caractéristiques sont souvent difficiles à satisfaire en même temps : les méthodes les plus robustes seront souvent plus lentes. Elles peuvent être également subjectives, suivant les problèmes auxquels on s'intéresse, on pourra par exemple privilégier la précision à la vitesse ou le contraire. Enfin on s'intéressera à des caractéristiques analogues (par exemple le coût total d'exécution, pas seulement en temps, mais peut-être en espace mémoire, etc.). ([2] : p4)

Ainsi, dans la première section nous allons commencer par expérimenter la méthode de la plus forte pente.

Dans la deuxième section la méthode de Newton.

Dans la troisième et dernière section nous allons établir une petite comparaison entre ces deux méthodes précitées.

Toutes les expérimentations s'appliqueront à une fonction à trois variables

$$f(x) = e^{x_1} + x_1^2 x_3 - x_1 x_2 x_3. \quad (4.1)$$

En voici alors l'essentiel :

4.1 Méthode de la plus forte pente

La table 4.1 présente les itérations de la fonction (4.1) ci-haut donnée.

En effet, en utilisant la méthode de la plus forte pente pour cette fonction, nous sommes parti d'un point initial $x_0 = (x_1, x_2, x_3) = (1, 1, 0.5)$ avec $\alpha_0 = 1$ à une tolérance $1e - 6$.

A chaque itération on calcule à la fois les valeurs de x_1, x_2, x_3 , la norme de $f(x)$ en ces trois points et enfin la norme du gradient de f . Les points successifs étant liés par la relation $x_{k+1} = x_k + \alpha_k d_k$ où x_k est la valeur de x à l'étape (itération) k et α_k en est le pas dans la direction d_k .

4.1.1 Tableau d'itérations

TABLE 4.1 – Itérations de (4.1) par la méthode de la plus forte pente

Itérations	x_1	x_2	x_3	$\ f(x)\ $	$\ \nabla f(x)\ $
1.00000	1.00000	1.00000	0.50000	3.71828	3.25689
1	0.99012	1.00154	0.50000	3.67754	3.21922
2	0.98024	1.00307	0.50004	3.63722	3.18188
3	0.97036	1.00461	0.50011	3.59730	3.14488
4	0.96048	1.00616	0.50021	3.55780	3.10821
5	0.95060	1.00770	0.50035	3.51869	3.07185
6	0.94072	1.00925	0.50053	3.47999	3.03582
7	0.93084	1.01080	0.50074	3.44167	3.00009
8	0.92097	1.01236	0.50099	3.40375	2.96467
9	0.91109	1.01391	0.50127	3.36621	2.92955
10	0.90122	1.01547	0.50159	3.32905	2.89472
11	0.89135	1.01703	0.50195	3.29226	2.86018
12	0.88148	1.01860	0.50234	3.25584	2.82593
13	0.87161	1.02016	0.50277	3.21978	2.79195
14	0.86175	1.02173	0.50323	3.18409	2.75825
15	0.85189	1.02331	0.50373	3.14875	2.72481
16	0.84203	1.02488	0.50427	3.11377	2.69163
17	0.83217	1.02646	0.50484	3.07913	2.65871
18	0.82231	1.02804	0.50545	3.04483	2.62603
19	0.81246	1.02962	0.50609	3.01087	2.59360
20	0.80261	1.03121	0.50677	2.97724	2.56141
30	0.70429	1.04720	0.51555	2.65837	2.25136
40	0.60639	1.06343	0.52783	2.36826	1.95893
50	0.50899	1.07984	0.54343	2.10308	1.67864
60	0.41213	1.09633	0.56202	1.85943	1.40533
70	0.31577	1.11273	0.58310	1.63433	1.13442
80	0.21970	1.12867	0.60586	1.42519	0.86220
90	0.12344	1.14330	0.62860	1.22967	0.58682
98	0.045438	1.152419	0.643749	1.081525	0.367753
99	0.035564	1.153214	0.645116	1.063479	0.341234

4.1.2 Graphique de la fonction

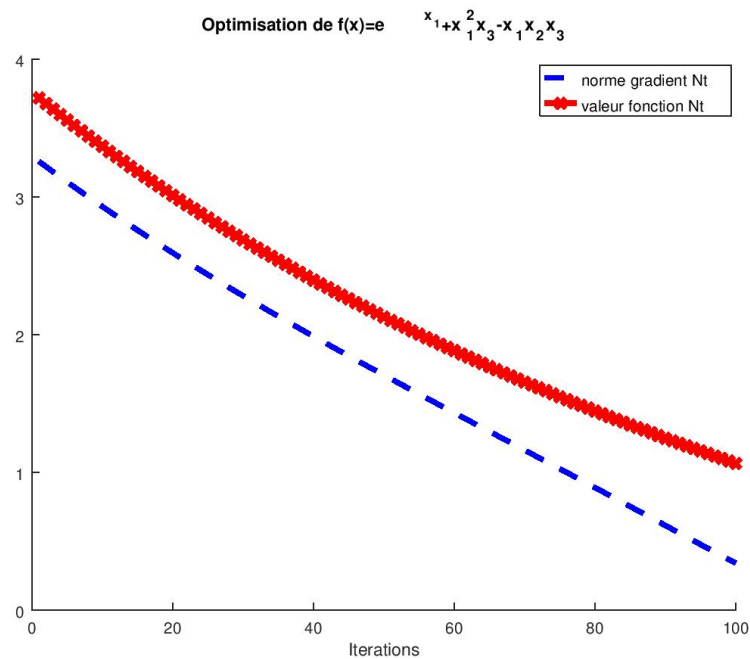


FIGURE 4.1 – Graphique associé à la fonction (4.1) par la méthode de la plus forte pente

Pour ce faire nous avons opté pour 100 comme nombre total d'itérations, et pour cela on constate que lorsqu'on s'approche de plus en plus de 100, la norme du gradient tend vers 0 et en vertu des conditions d'optimalité on peut dire aisément que la méthode converge vers un minimum local $x^* = (x_1, x_2, x_3) = (0.035564, 1.153214, 0.645116)$.

Voici alors le code octave qui a fournit à la fois les itérations ainsi que le graphique correspondant.

4.1.3 Code octave I

```
%
fctn = @(x) exp(x(1))+x(1)^2*x(3)-x(1)*x(2)*x(3);

g = @(x) [exp(x(1))+2*x(1)*x(3)-x(2)*x(3); -x(1)*x(3); x(1)^2-x(1)*x(2)];

h = @(x) [exp(x(1))+2*x(3), -x(3), 2*x(1)-x(2); -x(3), 0, -x(1); 2*x(1)-x(2), -x(1), 0];

maxIter = 100;
tol = 1e-6;
x0 = [1,1,0.5];
[X,normGrad,it]=optimFctn(fctn,g,h,x0,tol,maxIter);
[x] = descente_methOpt(fctn,g,x0,tol,maxIter);

[X,normGrad,it]=optimFctn(fctn,g,[],x0,tol,maxIter);
```

4.2 Méthode de Newton

En considérant toujours la fonction (4.1) au même point initial x_0 , nous présentons ci-après les itérations avec la méthode de Newton.

4.2.1 Tableau d'itérations

TABLE 4.2 – Itérations de (4.1) par la méthode Newton

Itérations	x_1	x_2	x_3	$\ f(x)\ $	$\ \nabla f(x)\ $
1.00000	1.00000	1.00000	0.50000	3.71828	3.25689
1	0.01	0.01	0.49	1.01	1.01
2	0.01	0.01	0.49	1.01	1.01
3	0.01	0.02	0.50	1.01	1.00
4	0.01	0.04	0.51	1.01	1.00
5	0.01	0.07	0.53	1.01	0.98
6	0.01	0.12	0.57	1.01	0.95
7	0.01	0.30	0.73	1.01	0.80
8	0.00	1.10	1.45	1.01	0.59
9	-0.00	0.85	1.24	1.00	0.05
10	-0.00	0.85	1.24	1.00	0.05
11	-0.00	0.85	1.24	1.00	0.05
12	-0.00	0.85	1.24	1.00	0.05

4.2.2 Graphique de la fonction

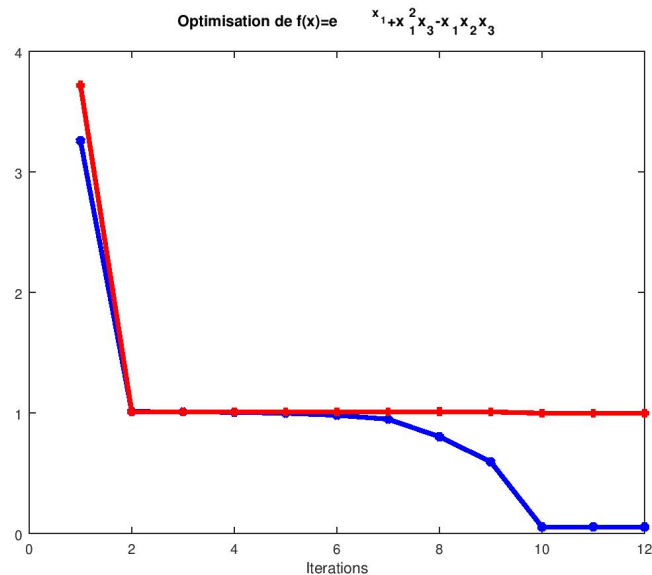


FIGURE 4.2 – Graphique de la fonction (4.1) par la méthode de Newton

Comme nous pouvons facilement le constater dans le tableau d'itérations tout comme au graphique, la méthode converge déjà vers la solution du problème à partir de la dixième itération.

Quant à la figure; la légende demeure la même que celle de la précédente figure.

Voici le code octave qui a généré les résultats présentés ci-haut pour la méthode de Newton :

4.2.3 Code octave II

```
%
fctn = @(x) exp(x(1))+x(1)^2*x(3)-x(1)*x(2)*x(3);

g = @(x) [exp(x(1))+2*x(1)*x(3)-x(2)*x(3); -x(1)*x(3); x(1)^2-x(1)*x(2)];

h = @(x) [exp(x(1))+2*x(3), -x(3), 2*x(1)-x(2); -x(3), 0, -x(1); 2*x(1)-x(2), -x(1), 0];

maxIter = 12;
tol = 1e-6;
x0 = [1,1,0.5];
%[X,normGrad,it]=optimFctn(fctn,g,h,x0,tol,maxIter);
%[x] = descente_methOpt(fctn,g,x0,tol,maxIter);

[X,normGrad,it]=optimFctn(fctn,g,h,x0,tol,maxIter);
```

4.3 Comparaison entre la méthode de Newton et celle de la plus forte pente

4.3.1 Graphiques des deux méthodes

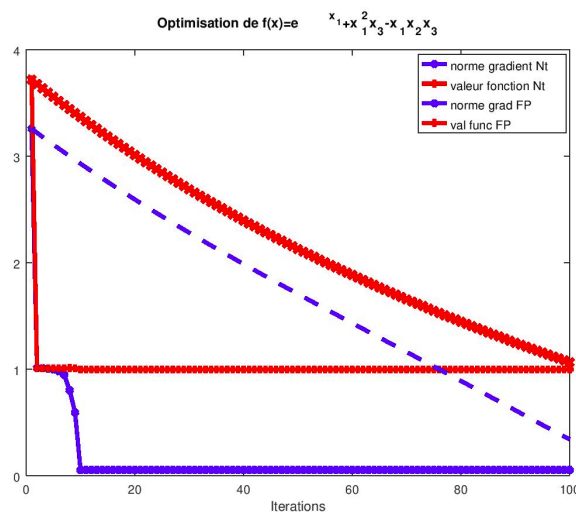


FIGURE 4.3 – Graphique de la Méthode de Newton et celui de la Plus forte pente

Cette figure 4.3 présente la comparaison entre la méthode de Newton et celle de la plus forte pente en utilisant toujours la fonction (4.1) au même point de départ x_0 .

Elle présente à la fois les variations de la valeur de la fonction et de son gradient en utilisant d'une part la méthode de la plus forte pente et d'autre part celle de Newton.

On remarque que la méthode de Newton a convergé plus rapidement vers la solution du problème par rapport à celle de la plus forte pente car déjà au tour de la 10e itération,

la norme du gradient est déjà nulle chez Newton quoi qu'il y ait stagnation quelque part alors que pour la plus forte pente c'est après la centième itération que la méthode tend à converger.

En vertu de ce qui précède ci-haut, nous en déduisons que la méthode de Newton serait plus rapide que celle de la plus forte pente, elle minimise donc à la fois le nombre de calculs et celui d'itérations contrairement à celle de la plus forte pente qui apparaît lente.

Voici alors le code octave qui a généré les résultats :

4.3.2 Code Octave III

```
%disp([0,x,fv,fd]);
Fv = [Fv;fv];
normGrad=[normGrad;norm(fd)];
it = 1;
disp([it, x,fv,norm(fd)]);
while (abs(fv) > tol && it < maxIter)
fd = fd./norm(fd);
x_old =x;
x = x_old-(alpha*fd');
fv = feval(f,x);
fd = feval(f_der,x); %eval('f_der',x);
Fv = [Fv;fv];
normGrad=[normGrad;norm(fd)];
disp([it,x,fv,norm(fd)]);
it = it+1;
end
hold on;
plot([1:it],normGrad(1:it),'--b','Linewidth',3);
hold on;
plot([1:it],Fv(1:it),'-xr','Linewidth',3);
title('Optimisation de f(x)=e^{x_1}+x_1^2x_3-x_1x_2x_3')
xlabel('Iterations')
legend('norme gradient Nt','valeur fonction Nt','norme grad FP','val func FP');
```

Synthèse partielle

Ce chapitre consistait à expérimenter juste un petit échantillon des méthodes précédemment présentées. Nous avons considéré une fonction donnée par (4.1) à laquelle nous avons appliqué deux méthodes dont celle de la plus forte pente et celle de Newton. Ces deux méthodes ont été implémentées dans le langage de programmation octave. Pour ce faire, nous avons opté pour 100 comme nombre total d'itérations en partant d'un point initial $x_0 = (x_1, x_2, x_3) = (1, 1, 0.5)$ avec $\alpha_0 = 1$ à une tolérance $1e - 6$. Il a été constaté avec la méthode de la plus forte pente que lorsqu'on s'approche de plus en plus de 100, la norme du gradient tend vers 0 et en vertu des conditions d'optimalité on peut dire aisément que la méthode converge vers un minimum local $x^* = (x_1, x_2, x_3) = (0.035564, 1.153214, 0.645116)$ alors que celle de Newton converge déjà vers la dixième itération.

Les résultats trouvés respectivement dans les tables 4.1 et 4.2 montrent que la méthode de Newton converge rapidement vers la solution du problème.

Conclusion

Nous voici au terme de ce travail ayant porté sur les méthodes numériques d'optimisation non linéaire. Premièrement nous avons procédé à la présentation des Généralités sur l'optimisation numérique où nous avons défini des notions jugées incontournables en vue d'une meilleure compréhension des diverses méthodes développées dans les deuxième et troisième chapitres.

Deuxièmement nous avons développé des méthodes dites d'optimisation sans contrainte qui sont effectivement des méthodes itératives conçues pour minimiser un problème d'optimisation non linéaire modélisé surtout par une fonction non linéaire. Nous avons tenté dans la mesure du possible de détailler suffisamment chaque méthode en présentant son objectif, ses exigences attendues de la part de l'utilisateur ainsi que le résultat susceptible qu'elle peut fournir.

Troisièmement nous avons parlé des méthodes d'optimisation avec contraintes; ce chapitre n'existerait pas sans surtout l'existence du deuxième car nous avons montré que des problèmes sous contraintes se réduisent généralement à ceux sans contraintes paraissant simples à résoudre, et donc par conséquent leurs méthodes de résolution sont basées surtout sur la même intuition presque que celle des méthodes sans contrainte. A ce niveau s'ajoute l'atteinte de deux objectifs pour la minimisation : diminuer la valeur de la fonction objectif et vérifier les contraintes. Néanmoins ces deux objectifs sont souvent en conflit, chose expliquant la complexité de ce chapitre par rapport au précédent. Cependant notre attention dans ce chapitre s'est surtout focalisée à des méthodes de résolution d'une fonction non linéaire soumise au moins à une contrainte linéaire.

Quatrièmement nous sommes passé aux applications numériques de certaines méthodes d'optimisation sans contrainte en implémentant quelques algorithmes dans le langage de programmation octave. Mais il est certes à signaler que ces algorithmes ont été choisis à volonté afin d'illustrer seulement certaines méthodes sans contraintes, notamment la méthode de la plus forte pente et celle de Newton.

Les résultats sus-cités traduisent que nos objectifs ont été atteints et constituent notre contribution dans ce travail. Néanmoins, nous ne prétendons pas avoir traité exhaustivement les méthodes numériques d'optimisation non linéaire. D'autres chercheurs pourraient nous emboîter le pas en faisant une comparaison systématique des diverses méthodes numériques d'optimisation non linéaire, une étude des différentes applications de ces méthodes dans un domaine scientifique donné ou tout simplement aussi l'étude des méthodes numériques d'optimisation à grande échelle.

Convaincu que toute œuvre humaine est imparfaite, les corrections et recommandations pour l'amélioration de celle-ci sont les bienvenues.

Bibliographie

- [1] AMANI FARAHA Déborah, *sur l'Optimisation multiobjectifs : Résolution du problème du voyageur de commerce*, Mémoire, Inédit, ISP-BUKAVU, 2018.
- [2] Amic Fouvelle, *Méthodes numériques : optimisation*, 2e édition, 18 Avril 2016.
- [3] Alfio Quarteroni et al., *Calcul scientifique : cours, exercices corrigés et illustrations en MATLAB et Octave*, Deuxième édition, Springer, Milan et Brescia, Mai 2010.
- [4] AUDE RONDE Pierre, *Méthodes numériques pour l'optimisation non linéaire déterministe*, Département génie mathématique et modélisation 4^e, Toulouse, 2017-2018.
- [5] BAHATI NZEYI Jean Claude, *Méthodes numériques de résolution des systèmes non linéaires*, TFC, Inédit, ISP-Bukavu, 2017.
- [6] BAHATI RUGAMBA Patient, *Méthodes d'optimisation sans contraintes pour la résolution des équations non linéaires*, TFC, Inédit, ISP-Bukavu, 2018.
- [7] B.P CHENITCHNY et Y.DANILINE, *Méthodes numériques dans les problèmes d'extremum*, Éditions MIR, Moscou, 1977.
- [8] Frédéric de Gournay & Aude Ronde Pierre, *Introduction à l'optimisation Numérique*, Département STPI, 3e année MIC, INSA, Toulouse, 2015.
- [9] Fulgence NAHAYO, *Modèle mathématique d'optimisation non linéaire du bruit des avions commerciaux en approche sous contrainte énergétique*, Thèse de l'université Claude Bernard, Lyon 1, 2012.
- [10] Jean Daniel BONJOUR, *Introduction à MATLAB et GNU Octave*, Section des sciences et ingénierie de l'environnement(SSIE), Faculté ENAC, Lausanne, 1999-2011.
- [11] JEDRZEJEWSKI, *Introduction aux méthodes numériques*, Deuxième édition, Springer, Paris, 2005.
- [12] LAURENT GUILOPE, *Optimisation sous contrainte*, Laboratoire de mathématiques Jean Leray, université de Nantes, Paris, 2017.
- [13] Mathias KLEINER, *Méthodes et outils d'optimisation : arts et métiers*, Mai 2013.
- [14] MICHEL BIERLAIRE, *Introduction à l'optimisation différentiable*, PPUR presses polytechniques, 2006.
- [15] M.Karim BENALIA, *Optimisation non linéaire avec contraintes, méthodes d'optimisation méta heuristiques et problème de routage de véhicules : Application aux problèmes pétroliers : cas de l'entreprise SONATRACH*, Université MOULOUD MAMMERI, 2014.
- [16] S.LEDIGABEL, *Optimisation non linéaire : Théorie*, Polytechnique Montréal, 2019.
- [17] SORIN CIUPERCA, *Cours Optimisation*, cours en master M1 SITN, inédit, 2018.
- [18] Stephane MOTTELET, *Optimisation non linéaire*, Université de Technologie de compiègne, Printemps 2003.
- [19] WALTER, *Méthodes numériques et optimisation :un guide du consommateur*, Université Paris-sud, 2015