

Titre: Optimisation riemannienne sans dérivées pour les contraintes d'égalité
Title:

Auteur: Sacha Benarroch
Author:

Date: 2026

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Benarroch, S. (2026). Optimisation riemannienne sans dérivées pour les contraintes d'égalité [Master's thesis, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/76210/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/76210/>
PolyPublie URL:

Directeurs de recherche: Charles Audet, & Sébastien Le Digabel
Advisors:

Programme: Maîtrise recherche en mathématiques appliquées
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Optimisation riemannienne sans dérivées pour les contraintes d'égalité

SACHA BENARROCH

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques

Avril 2026

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Optimisation riemannienne sans dérivées pour les contraintes d'égalité

présenté par **Sacha BENARROCH**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Serge PRUDHOMME, président

Charles AUDET, membre et directeur de recherche

Sébastien LE DIGABEL, membre et codirecteur de recherche

Youssef DIOUANE, membre

DÉDICACE

*À celles et à ceux qui cultivent
l'art de humer les pins le long du chemin.*

REMERCIEMENTS

Entrer dans le monde de la recherche scientifique est une chose dont l'étudiant ingénieur que j'étais a rêvé. Cher Charles, cher Sébastien, cet étudiant vous est reconnaissant de lui avoir ouvert votre porte, et d'avoir accepté de lui mettre le pied à l'étrier. Celui qu'il est devenu constate ce que la qualité de votre encadrement lui a apporté, mais également sa manie de parler de lui à la troisième personne (non pas que cela doive vous être imputé). Votre curiosité, votre art de guider sans imposer, d'écouter tout en critiquant, ont considérablement affecté mes apprentissages de ces deux années, ainsi que les pages qui suivent. Parmi les découvertes que nous avons faites ensemble, une me marque particulièrement : en plus d'adoucir les mœurs, la musique a adouci quelques unes des migraines que nous avons pu traverser durant nos rencontres. De ces rencontres, je garderai le souvenir de la bonne humeur constante et de la bienveillance. La confiance que vous m'avez témoignée durant ces deux ans m'honore : d'une part, en me laissant explorer un domaine de recherche hors de votre zone de confort ; d'une autre, en me permettant de vérifier à plusieurs reprises que l'enseignement figure effectivement parmi les plus belles activités qui soient. Pour tout ce que j'ai appris durant ces deux années, ma gratitude est grande. Pourvu que l'avenir nous donne l'occasion de collaborer de nouveau.

La recherche scientifique est une entreprise souvent solitaire, mais certains savent faire d'un labo bien plus qu'un lieu où les solitudes se juxtaposent. Aux voisins et amis du GERAD : Maxence, Adrien, Xavier, Léo, Vincent, Théo, Nathan, Mathieu, Camille, Edward, Valentin, Mayssem, un très grand merci. Merci pour les longues discussions, tantôt trop scientifiques et tantôt très peu, pour les repas et les verres partagés. Tout cela contribue à faire avancer les idées, et à reconstruire un chez-soi quand on en est loin. Cela va aussi à celles et ceux que j'ai rencontrés grâce à vous : à Léa, à Saige, à Yali, et j'en oublie.

Ce mémoire et son auteur (réellement déterminé à parler de lui à la troisième personne) doivent beaucoup à un collègue et ami en particulier. Pierre-Yves, je ne suis pas en mesure de te rendre tous les après-midis que tu as consacrés à nos discussions, mais il faut noter que si ces moments eussent été plus courts, la face de ce mémoire aurait probablement changé. Ils ont été remplis d'idées précieuses pour mon travail et de rire (dans des proportions variables), j'en garde d'excellents souvenirs.

À toi Georges qui, après m'avoir transmis l'amour de la musique, m'avais dit que je l'emmènerais partout avec moi : tu avais raison. Grâce à toi et à cet amour, j'ai fait de merveilleuses rencontres. Merci à l'Orchestre Symphonique Étudiant de l'Université de Montréal, qui a été mon apport hebdomadaire de sourires et de belle musique. Jonathan, Laurence, Timon,

Anaïs, Sandrine, je m'estime chanceux d'être lié à vous de cette façon.

Aux cinq classes auxquelles j'ai eu le privilège d'enseigner l'algèbre linéaire à Polytechnique, merci pour tout ce que vous m'avez appris sur ce si beau métier.

J'adresse mes plus sincères remerciements au secteur *Nature et Technologies* des Fonds de Recherche du Québec, dont le programme de bourse de maîtrise en recherche a financé une partie de ce travail.

Enfin, à mes deux familles de France. À celle que l'on se choisit, en particulier Quentin, Alexandre, Marie, Mauricio, Tizziana : vos amitiés sont des cadeaux précieux. Et à celle que l'on ne choisit pas, mais qui nous rend heureux que le choix ne fût pas laissé. À mon frère Nicolas, qui n'est plus le petit frère que par l'âge, et qui me rend un fier grand frère. Et à ma Maman, dont l'amour et le courage sans limite m'ont donné les ressources pour m'accomplir. Je vous aime autant qu'il est possible.

RÉSUMÉ

Cette recherche s'intéresse à la résolution de problèmes d'optimisation sans dérivées dans lesquels interviennent des contraintes d'égalité. Elle est motivée par le constat suivant : dans la littérature, certaines méthodes traitent ces problèmes en évaluant une suite de points ne respectant pas ces contraintes, jusqu'à converger vers une solution satisfaisante à la fois du point de vue des contraintes et de celui de l'objectif. Cette approche n'est pas toujours envisageable dans le cadre de l'optimisation sans dérivées : puisque la fonction que le problème cherche à optimiser peut correspondre à une simulation informatique, il est possible qu'aucune valeur ne puisse être obtenue à moins de respecter tout ou partie des contraintes.

La question qui se pose alors est la suivante : si certaines contraintes sont connues et doivent être respectées par chaque itéré, comment exploiter cette connaissance pour guider un algorithme d'optimisation sans dérivées à travers le domaine réalisable ? Une des réponses à cette question se situe dans l'étude de la géométrie de ce domaine. Un algorithme d'optimisation sans dérivées est généralement incapable de suivre un chemin tracé par un tel domaine dans $\mathbb{R}^n$: il doit pouvoir se déplacer en ligne droite dans n'importe quelle direction de l'espace et obtenir l'information qui lui permet d'inférer la prochaine direction à explorer. Une solution à ce problème peut être apportée dans le cas où le domaine est suffisamment régulier pour admettre une linéarisation locale en chacun de ses points. Un tel ensemble est appelé une *sous-variété riemannienne* de $\mathbb{R}^n$. Cependant, exécuter un algorithme sur une linéarisation locale ne fournit pas des points satisfaisant exactement les contraintes d'égalité. Pour pallier ce problème, nous utilisons la notion de *rétraction* issue de l'optimisation riemannienne.

Ce mémoire s'intéresse donc au cas où l'ensemble $\Omega_=_$ réalisable pour les contraintes d'égalité d'un problème constitue une sous-variété riemannienne de $\mathbb{R}^n$, et traduit cette condition directement sur les contraintes concernées. Il propose une méthode nommée RDFO (pour *Riemannian Derivative-Free Optimization*), capable d'exécuter un solveur sans dérivées sur les espaces dits *tangents* à $\Omega_=_$, tout en conservant un contrôle sur l'exactitude avec laquelle sa linéarisation le décrit. Dans le cas où le solveur en question est l'algorithme MADS (*Mesh Adaptive Direct Search*), nous montrons que certaines conditions nécessaires d'optimalité sont obtenues lorsque cette méthode converge.

Enfin, des expériences numériques sont menées, afin d'évaluer les capacités de la méthode sur diverses familles de problèmes.

ABSTRACT

This research is interested in solving equality-constrained derivative-free optimization problems. It is motivated by the fact that most methods within the literature solve such a problem by evaluating a sequence of so-called *nonfeasible points*, i.e., points that do not satisfy those constraints, finally aiming at a solution that is both feasible for the constraints and improving for the objective function. This setup cannot always be used in the realm of derivative-free optimization. Indeed, evaluating the objective function quite often results from a computer simulation that might fail in case one or more of the constraints is/are not satisfied.

If a user knows about these so-called *unrelaxable* constraints, how can this knowledge be used to guide a derivative-free optimization algorithm throughout the set that satisfies these constraints? A way to answer this question is to look at the geometry of the set of feasible points for these constraints. A derivative-free optimization algorithm is typically unable to follow the curved path defined by such a subset : it needs to be able to move in straight lines and gather the objective value information that guides it towards its next iterate. A solution to this problem exists when this set is regular enough so it can be locally linearized close to each of its points. Such a subset is called a *Riemannian submanifold* of $\mathbb{R}^n$. However, running a derivative-free solver on such a local linearization does not yield perfectly feasible points. Although projection is a workaround, we seek to gain more thorough depiction of the links between the set of feasible points and its linearizations. We thus resort to concepts from differential geometry.

This thesis is thus interested in the case where the set $\Omega_=_$ of feasible points for the equality constraints is such a locally linear set. It translates this on explicit conditions on the equality constraints. A feasible method named RDFO (*Riemannian Derivative-Free Optimization*) is introduced. This method is capable of running a derivative-free solver on the so-called *tangent spaces* to $\Omega_=_$ while keeping track of how precise the mapping between the tangent space and the original set is. In the case where the MADS (*Mesh Adaptive Direct Search*) solver is used, we show that convergence of this method leads to some necessary optimality conditions.

Finally, numerical experiments are presented in order to assess the method's abilities on various problem sets.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	vi
ABSTRACT	vii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES NOTATIONS ET ABRÉVIATIONS	xiii
CHAPITRE 1 INTRODUCTION	1
1.1 Historique de l'optimisation mathématique	1
1.2 Domaine de l'optimisation sans dérivées	2
1.3 Cadre et objectifs de la recherche	2
1.4 Démarche scientifique	4
1.5 Organisation du mémoire	5
CHAPITRE 2 ÉTAT DE L'ART EN OPTIMISATION SANS DÉRIVÉES	
SOUS CONTRAINTES D'ÉGALITÉ	7
2.1 Optimisation sans dérivées sous contraintes	7
2.1.1 Recherches directes directionnelles, algorithme MADS	7
2.1.2 Analyse de convergence de MADS	11
2.1.3 Traitement des problèmes contraints avec MADS	13
2.1.4 Limitations en présence d'égalités	15
2.2 Méthodes non réalisables	16
2.2.1 Interpolation, gradients et hessiennes simples	17
2.2.2 Modèles linéaires : COBYLA	17
2.2.3 Modèles quadratiques : SQP	19
2.2.4 Méthodes de pénalité et de barrière	20
2.2.5 Méthodes de restauration inexacte	21
2.3 Méthodes Ω_- -réalisables	22
2.3.1 Méthodes entièrement réalisables	22
2.3.2 Cas des égalités linéaires	22

2.4	Comparaison numérique d'algorithmes d'optimisation	23
CHAPITRE 3 Outils d'optimisation riemannienne		26
3.1	Sous-variétés riemanniennes de $\mathbb{R}^n$	26
3.2	Espaces tangents	30
3.3	Revenir sur Terre : rétractions	31
3.3.1	Pourquoi projeter ne suffit pas	32
3.3.2	Rétractions, géodésiques et application exponentielle	32
3.3.3	Une projection peut être une rétraction	35
3.4	Algorithmes d'optimisation sur variétés	36
CHAPITRE 4 Conception d'une méthode Ω_- -réalisable sur variétés riemanniennes		38
4.1	Principes et outils de RDFO	38
4.1.1	Sous-problèmes dans $\mathcal{TM}$	39
4.1.2	Rétractions et rayon d'inversibilité	39
4.1.3	Idée et notations de RDFO	41
4.2	L'algorithme	41
4.3	Analyse de convergence de RDFO avec MADS	43
4.3.1	Convergence de MADS sur le dernier sous-problème	43
4.3.2	Opération de transfert	44
4.3.3	Résultat de convergence	47
CHAPITRE 5 Implémentation et résultats numériques		51
5.1	Implémentation	51
5.1.1	Reformulation des sous-problèmes pour l'implémentation	52
5.1.2	Bases des espaces tangents	52
5.1.3	Rétractions et rayons d'inversibilité	53
5.1.4	Traitement des contraintes d'inégalité	54
5.2	Première illustration	54
5.3	Contraintes d'égalité linéaires	55
5.3.1	Ensemble de problèmes à contraintes d'égalités linéaires	57
5.3.2	Comparaison à la méthode des convertisseurs linéaires	57
5.4	Problèmes définis sur des sphères	58
5.4.1	Problèmes de valeurs propres extrêmes	59
5.4.2	Ensemble de problèmes définis sur des sphères	60
5.4.3	Choix d'une rétraction et d'une borne sur le rayon d'inversibilité	60

5.4.4	Comparaison à d'autres implémentations de MADS	63
5.5	Contraintes d'égalité non linéaires	64
5.5.1	Banque de problèmes	64
5.5.2	Résultats comparatifs	64
CHAPITRE 6	CONCLUSION	72
6.1	Synthèse des contributions	72
6.2	Limitations et perspectives de poursuite des travaux	73
RÉFÉRENCES		75

LISTE DES TABLEAUX

Tableau 5.1	Constitution du jeu de problèmes test pour les contraintes d'égalité linéaires	57
Tableau 5.2	Constitution du jeu de problèmes test sur des sphères . .	61
Tableau 5.3	Constitution du jeu de problèmes test pour les contraintes d'égalité non linéaires	64

LISTE DES FIGURES

Figure 3.1	Variété, ou non ?	29
Figure 3.2	Deux rétractions distinctes sur $\mathbb{S}^2$	34
Figure 4.1	Illustration des suites impliquées dans la preuve du théo- rème 4.4	47
Figure 5.1	Illustration du comportement de RDFO sur le problème (5.3)	56
Figure 5.2	Profils de données sur l'ensemble de problèmes test à contraintes d'égalité linéaires	66
Figure 5.3	Résolution du problème (5.5) de plus petite valeur propre par RDFO	67
Figure 5.4	Comportement de RDFO avec différentes rétractions et rayons d'inversibilité sur le problème (5.3)	68
Figure 5.5	Profils de données comparant les différentes rétractions et les rayons d'inversibilité associés sur l'ensemble de pro- blèmes test définis sur des sphères	69
Figure 5.6	Profils de données sur l'ensemble de problèmes test définis sur des sphères	70
Figure 5.7	Profils de données sur l'ensemble de problèmes test à contraintes d'égalité non linéaires	71

LISTE DES NOTATIONS ET ABRÉVIATIONS

Notations mathématiques

$\mathbb{R}^{m \times n}$	Ensemble des matrices à m lignes et n colonnes, à coefficients réels
$\text{rang}(A)$	Rang d'une matrice $A \in \mathbb{R}^{m \times n}$
$\text{Ker}(A)$	Noyau de la matrice $A \in \mathbb{R}^{m \times n}$
$\text{Sp}(A)$	Spectre (ensemble des valeurs propres) d'une matrice $A \in \mathbb{R}^{n \times n}$
$\lambda(A)$	Rayon spectral d'une matrice $A \in \mathbb{R}^{n \times n}$
$ E $	Cardinalité d'un ensemble E
$\text{int}(E)$	Intérieur d'un ensemble E
proj_E	Opérateur de projection métrique sur un ensemble E
$\{x\} + E$	Somme de Minkowski : $\{x + e : e \in E\}$
$\Omega_{\leq}, \Omega_{=}$	Ensembles réalisables pour les contraintes d'inégalité et d'égalité d'un problème d'optimisation
Ω	Ensemble réalisable pour un problème d'optimisation
$T_{\Omega}^H(x)$	Cône hypertangent à l'ensemble Ω au point $x \in \Omega$
$f'(x; d)$	Dérivée directionnelle (de Dini) de f au point x dans la direction d
$f^{\circ}(x; d)$	Dérivée directionnelle généralisée de Clarke de f au point x dans la direction d
$J_f(x)$	Matrice jacobienne de la fonction f évaluée en x
$H_f(x)$	Matrice hessienne de la fonction f évaluée en x
$\mathcal{C}^k$	Classe des fonctions k fois continûment différentiables
$\mathcal{M}$	Sous-variété riemannienne de $\mathbb{R}^n$
$\dim(F), \dim(\mathcal{M})$	Dimension du sous-espace vectoriel F de $\mathbb{R}^n$, dimension de la sous-variété riemannienne $\mathcal{M}$
$T_x \mathcal{M}$	Espace tangent à la sous-variété $\mathcal{M}$ en $x \in \mathcal{M}$
$T\mathcal{M}$	Fibré tangent de $\mathcal{M}$
$\langle \cdot, \cdot \rangle$	Produit scalaire canonique de $\mathbb{R}^n$
$\langle \cdot, \cdot \rangle_x$	Produit scalaire sur $T_x \mathcal{M}$
$\ \cdot\ $	Norme euclidienne dans $\mathbb{R}^n$
$\ \cdot\ _x$	Norme induite par le produit scalaire sur $T_x \mathcal{M}$
$B(x, \varepsilon)$	Boule ouverte de centre x et de rayon ε : $\{y \in \mathbb{R}^n : \ y - x\ < \varepsilon\}$
$B_x(v, \varepsilon)$	Boule ouverte de la topologie induite sur $T_x \mathcal{M}$: $\{w \in T_x \mathcal{M} : \ w - v\ < \varepsilon\}$

R_x	Rétraction sur $\mathcal{M}$ en x
Exp_x	Application exponentielle sur $\mathcal{M}$ en x
$\text{inj}(x)$	Rayon d'injectivité de l'application exponentielle en $x \in \mathcal{M}$
$\text{inv}(R_x)$	Rayon d'inversibilité de la rétraction R_x
$\rho(R_x)$	Borne inférieure sur le rayon d'inversibilité de R_x
$\pi_{\mathcal{B}}$	Plongement dans $T_x\mathcal{M}$ associé à une base $\mathcal{B}$ de $T_x\mathcal{M}$
$\mathbb{S}^{n-1}$	Sphère unité de $\mathbb{R}^n$

Abréviations

DFO	<i>Derivative-Free Optimization</i>
RDD	Recherche Directe Directionnelle
Cs	<i>Coordinate Search</i>
GPS	<i>Generalized Pattern Search</i>
MADS	<i>Mesh Adaptive Direct Search</i>
RDFO	<i>Riemannian Derivative-Free Optimization</i>
COBYLA	<i>Constrained Optimization BY Linear Approximation</i>

CHAPITRE 1 INTRODUCTION

Dans le langage mathématique moderne, une tâche d'*optimisation* se formule comme suit : étant donné un ensemble E et une fonction $f : E \rightarrow \mathbb{R} \cup \{\pm\infty\}$, identifier un élément $\hat{x} \in E$ tel que pour tout $y \in E$, $f(\hat{x}) \leq f(y)$. Dans le cas où un tel $\hat{x}$ existe, ce problème est bien souvent résumé par la notation

$$\min_{x \in E} f(x).$$

E est alors appelé l'*ensemble réalisable* du problème d'optimisation et f , sa *fonction objectif*. Les caractéristiques de l'ensemble E et de la fonction f déterminent la ou les méthodes qui peuvent être utilisées pour identifier un tel $\hat{x}$, qui est appelé *solution* du problème d'optimisation.

1.1 Historique de l'optimisation mathématique

Il n'a pas fallu attendre le développement du langage mathématique que nous connaissons aujourd'hui pour que les problèmes d'optimisation voient le jour dans les sciences. La première instance d'un problème d'optimisation dans la littérature scientifique est généralement attribuée à Héron d'Alexandrie (I^{er} siècle de notre ère), qui étudia dans sa *Catoptrique* la réflexion de la lumière en s'appuyant sur le concept de *plus court chemin* [23].

C'est la formalisation du calcul différentiel par Newton et Leibniz au XVII^{ème} siècle qui fournit à l'optimisation un moyen d'essor : la capacité d'étudier analytiquement les variations d'une fonction par ses dérivées. La méthode de Newton ramène alors un problème d'optimisation à un problème d'identification de racines des dérivées de f . De là naissent plusieurs domaines d'études, dont le calcul des variations motivé par le problème de la *brachistochrone*, posé par Bernoulli en 1696.

L'optimisation mathématique telle qu'elle est majoritairement pratiquée aujourd'hui naît au moment de la Seconde Guerre mondiale. Les travaux de George Dantzig sur l'allocation des ressources militaires laissent entrevoir de nombreuses applications. En résulte un grand nombre de disciplines : optimisation linéaire, en nombres entiers, quadratique, semi-définie, stochastique, théorie des jeux, et tant d'autres.

La discipline de l'optimisation à laquelle s'intéresse ce mémoire est singulière : il s'agit de celle qui traite un problème dans lequel les dérivées de la fonction objectif (et potentiellement, des contraintes définissant l'ensemble E) sont inaccessibles, voire inexistantes.

1.2 Domaine de l'optimisation sans dérivées

Dans de nombreuses applications de l'optimisation mathématique, il arrive que les quantités à optimiser ne soient pas exprimables de manière analytique, et que leurs dérivées ne soient pas accessibles, ou le soient à un coût élevé. C'est ici qu'apparaît l'intérêt de déporter nos efforts sur une exploration intelligente de l'espace des variables, en renonçant totalement à l'utilisation d'outils analytiques sur les fonctions à optimiser. L'optimisation sans dérivées (*Derivative-Free Optimization*, abrégée DFO dans ce mémoire) est ainsi *l'étude mathématique des algorithmes d'optimisation qui n'utilisent pas de dérivées* [15, 31, 68].

En pratique, l'optimisation sans dérivées intervient souvent dans le cas où la fonction à minimiser est un retour de calculs informatiques lourds, dont on ne peut (ou ne veut) pas modéliser le comportement de façon analytique. Les variables du problème sont alors les entrées fournies au programme qui effectue ces calculs, et ce programme est appelé une *boîte noire*. Pour cette raison, on appelle également cette discipline *optimisation de boîtes noires* (*Blackbox optimization*). Dans la pratique, puisque les calculs nécessaires à évaluer la boîte noire sont coûteux, un objectif central des méthodes d'optimisation sans dérivées est d'identifier des solutions en un nombre aussi faible que possible d'évaluations [46, 32].

Les applications connues de l'optimisation sans dérivées sont nombreuses, allant de la conception aéronautique au choix d'hyperparamètres en apprentissage machine, en passant par l'astrophysique ou encore la planification de la production électrique, pour n'en citer que peu [9, 10].

1.3 Cadre et objectifs de la recherche

Ce mémoire s'intéresse à une catégorie vaste de problèmes d'optimisation sans dérivées, dans lesquels des solutions sont recherchées à l'intérieur d'un domaine défini par un ensemble d'inégalités et d'égalités, appelées les *contraintes* du problème. Nous définissons précisément ici les caractéristiques des problèmes qui nous occupent.

Le problème d'intérêt dans le cadre de ce mémoire est le suivant :

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimiser}} && f(x) \\ & \text{sous contraintes} && g(x) \leq 0 \\ & && h(x) = 0 \end{aligned} \tag{P}$$

où :

- ▷ $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction appelée *fonction objectif* du problème ;
- ▷ $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ et chaque composante $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ de g est une fonction appelée $i^{\text{ième}}$ *contrainte d'inégalité* du problème ($i \in \llbracket 1, m \rrbracket$) ;
- ▷ $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ et chaque composante h_j de h est une fonction appelée $j^{\text{ième}}$ *contrainte d'égalité* du problème ($j \in \llbracket 1, p \rrbracket$).

Les égalités $h(x) = 0$ et inégalités $g(x) \leq 0$ doivent donc être comprises composante par composante.

Diverses méthodes sont étudiées pour résoudre le problème (P). Elles partagent toutes l'hypothèse minimale suivante : *les formes analytiques ainsi que les gradients des fonctions f et g sont inaccessibles*. Autrement dit, l'aspect *sans dérivées* des problèmes abordés est toujours porté au moins par l'objectif et les contraintes d'inégalité. Certaines méthodes supposent que les formes analytiques et/ou les gradients de h sont disponibles, d'autres non.

Il est à noter qu'aucune hypothèse n'est faite à ce stade sur les natures des fonctions f , g et h . De telles hypothèses, notamment sur leurs continuités et classes de différentiabilité interviennent dans ce mémoire, et sont précisées en temps voulu.

Les conventions suivantes sont de plus retenues :

- ▷ n est un entier naturel non nul appelé la *dimension* de (P) ;
- ▷ m est un entier naturel quelconque ;
- ▷ p est un entier naturel non nul, de sorte que les problèmes d'intérêt dans ce mémoire sont tous contraints par au moins une égalité ;
- ▷ ces trois lettres sont utilisées à travers le mémoire sans plus de précision pour désigner respectivement la dimension d'un problème d'optimisation, son nombre de contraintes d'inégalités et son nombre de contraintes d'égalités. Ainsi, on s'autorise à écrire par exemple « $n = 4, m = 2, p = 1$ » pour désigner une instance de (P) à quatre variables, contrainte par deux inégalités et une égalité.
- ▷ on appelle *ensemble réalisable pour les inégalités* le sous-ensemble de $\mathbb{R}^n$ défini par

$$\Omega_{\leq} \triangleq \{x \in \mathbb{R}^n : g(x) \leq 0\}.$$

De façon analogue, on appelle *ensemble réalisable pour les égalités* l'ensemble

$$\Omega_{=} \triangleq \{x \in \mathbb{R}^n : h(x) = 0\}.$$

Enfin, l'ensemble réalisable pour (P) est noté $\Omega \triangleq \Omega_{\leq} \cap \Omega_{=}$.

Ce mémoire s'intéresse principalement à des méthodes offrant des *garanties théoriques de*

convergence pour la résolution de (P). La revue de littérature du chapitre 2 évoque des méthodes dites *heuristiques*, mais elles ne sont pas le centre d'intérêt de ce travail.

De manière générale, l'optimisation sous contraintes est abordée sous deux angles. D'une part, les méthodes dites *non réalisables* génèrent itérativement des suites de points de $\mathbb{R}^n$ dont certains en dehors de Ω , puis utilisent des mesures de la violation des contraintes pour guider l'optimisation vers des points à la fois réalisables, et améliorant la valeur de l'objectif. Cette approche n'est pas toujours envisageable dans un contexte d'optimisation de boîte noire, dans lequel il est possible que tout ou partie des contraintes doivent être satisfaites par un point pour qu'il soit possible d'obtenir les valeurs associées de l'objectif et des autres contraintes. Ceci est notamment possible lorsque certaines fonctions de (P) sont données par une simulation informatique, dans laquelle un itéré devrait respecter certaines contraintes pour pouvoir lancer une simulation. Ces contraintes peuvent être imposées par la signification physique des variables, ou dans le cas où les variables étudiées forment une distribution de probabilité, par exemple. Une contrainte dont la violation empêche d'évaluer les autres fonctions de (P) est appelée *non relaxable* pour (P). À l'inverse, les méthodes dites *réalisables* génèrent des suites de points incluses dans Ω , et peuvent donc être d'une utilité cruciale en DFO.

Ce mémoire s'intéresse au cas où un solveur sans dérivées doit résoudre un problème du type de (P) pour lequel *les contraintes d'égalité sont connues et non relaxables*. L'objectif principal est de concevoir une méthode d'optimisation sans dérivées capable d'utiliser la connaissance des contraintes d'égalité pour générer une suite d'itérés comprise non pas dans Ω , mais dans Ω_- . Nous appelons une telle approche Ω_- -réalisable. En particulier, nous cherchons à ce que cette méthode puisse être utilisée avec l'algorithme *Mesh Adaptive Direct Search* (MADS) présenté à la section 2.1.1 en tirant profit de ses garanties théoriques de convergence. Ces garanties sont sujettes à de fortes limitations en présence de contraintes d'égalité, la méthode conçue vise à les pallier.

1.4 Démarche scientifique

Plusieurs stratégies sont développées dans la littérature pour résoudre des problèmes contraints avec l'algorithme MADS. Ces stratégies sont détaillées dans la section 2.1.3, elles sont souvent utilisées en présence de contraintes d'inégalité mais échouent généralement en présence de contraintes d'égalité. Cet échec est lié à la géométrie de l'ensemble Ω_- , qui est d'intérieur vide dans la plupart des cas.

Or si les contraintes d'égalité du problème à résoudre sont connues, il est possible de prédire

cette géométrie, et de l'exploiter. Sous certaines hypothèses sur h , l'ensemble Ω_+ vérifie une propriété très utile pour travailler avec des algorithmes du type de MADS : il forme une *sous-variété riemannienne* de $\mathbb{R}^n$. Le chapitre 3 définit précisément ce que signifie ce terme ; de façon imprécise, on peut dire qu'il s'agit d'un sous-ensemble de $\mathbb{R}^n$ suffisamment régulier pour être linéarisé près de chacun de ses points. La méthode développée dans ce mémoire formule une série de sous-problèmes définis sur des linéarisations locales de Ω_+ . Puisque ces linéarisations sont des sous-espaces euclidiens de $\mathbb{R}^n$, résoudre de tels sous-problèmes avec un algorithme comme MADS fournit des garanties théoriques de convergence. L'objectif est de transférer ces garanties théoriques depuis les sous-problèmes vers le problème général.

La méthode obtenue, appelée RDFO (pour *Riemannian Derivative-Free Optimization*), est capable de résoudre différents types de problèmes, intersectant les capacités de plusieurs autres solveurs de la littérature (dédiés aux égalités linéaires, aux problèmes d'optimisation riemannienne...), Ω_+ -réalisables ou non. Plusieurs rondes de tests sont donc menées afin de comparer RDFO à d'autres solveurs, sur des ensembles de problèmes spécifiques aux capacités de chacun de ces solveurs.

1.5 Organisation du mémoire

Ce mémoire allie deux domaines de l'optimisation mathématique : l'optimisation sans dérivées et l'optimisation riemannienne (ou optimisation sur variétés). Le chapitre 2 propose une revue de la littérature exclusivement dédiée à l'optimisation sans dérivées sous contraintes d'égalité, en évoquant à la fois les méthodes non réalisables et les méthodes réalisables, mais laisse entièrement de côté tout ce qui a trait à l'optimisation riemannienne.

Le chapitre 3 est dédié à l'optimisation riemannienne mais il ne se veut pas une revue de sa littérature. Il introduit des outils choisis, en restant dans la perspective de construction de la méthode abordée au chapitre 4. Certains résultats qui ne sont pas essentiels en optimisation riemannienne mais utiles ici sont ainsi mis en avant, ou parfois reformulés puis démontrés pour satisfaire les besoins de ce mémoire, tandis que d'autres sont mis de côté.

Le cœur des contributions de ce mémoire est présenté dans les chapitres 4 et 5. Le chapitre 4 utilise les outils du chapitre 3 pour introduire la méthode que nous proposons, nommée RDFO (pour *Riemannian Derivative-Free Optimization*). Il en détaille le fonctionnement, et propose une analyse de ses garanties théoriques de convergence. Le chapitre 5 s'intéresse à l'implémentation de cette méthode, notamment à la reformulation des sous-problèmes et au choix de certains paramètres. Une série de tests numériques est menée afin de choisir des valeurs pour ces paramètres. D'autres tests sont également présentés pour illustrer les per-

formances de RDFO face à d'autres algorithmes, sur trois ensembles de problèmes contraints par des égalités de natures diverses. Ces tests soulignent la versatilité de RDFO.

Remarque sur l'utilisation des outils d'intelligence artificielle. Durant le travail de recherche qui a mené à l'écriture de ce mémoire, une utilisation limitée et contrôlée a été faite des outils d'intelligence artificielle générative (IAG). Leur utilisation a visé à la reformulation de certains concepts mathématiques, à confronter des raisonnements mathématiques et à orienter précisément la revue de littérature par moments. Sur le plan logiciel, l'utilisation d'outils d'IAG a servi à assister le développement de façon mineure. Sur le plan de la rédaction, des outils d'IAG ont été utilisés principalement pour aider à la construction des figures présentes dans ce mémoire. L'auteur déclare que le texte présenté ici est entièrement son œuvre, et qu'aucun résultat ni raisonnement ne provient d'une IAG.

CHAPITRE 2 ÉTAT DE L'ART EN OPTIMISATION SANS DÉRIVÉES SOUS CONTRAINTES D'ÉGALITÉ

Ce mémoire s'intéresse à l'optimisation sous contraintes d'égalité à travers le prisme de l'optimisation riemannienne. Afin de ne pas surcharger le lecteur en détaillant toutes les notions de géométrie différentielle qui y sont reliées, nous proposons dans ce chapitre une revue de la littérature concernant les méthodes d'optimisation sans dérivées sous contraintes d'égalité qui ne font pas appel à l'optimisation riemannienne. Tout ce qui concerne cette dernière est présenté au chapitre 3.

Nous introduisons d'abord une famille de méthodes d'optimisation sans dérivées appelées *recherches directes directionnelles*, et montrons leurs limitations en présence de contraintes d'égalité. Nous abordons ensuite la façon dont l'ensemble de la littérature de l'optimisation sans dérivées traite les problèmes présentant des contraintes d'égalité. Deux approches sont distinguées : l'approche *non réalisable*, qui évalue des points ne satisfaisant pas les contraintes du problème, et l'approche Ω_- -*réalisable*, qui cherche à ne générer que des points satisfaisant au minimum les contraintes d'égalité. Nous terminons en évoquant un outil transversal de l'optimisation sans dérivées utilisé dans ce mémoire pour comparer les performances numériques d'algorithmes.

2.1 Optimisation sans dérivées sous contraintes

Cette section est dédiée à l'introduction d'un type particulier de méthodes d'optimisation sans dérivées : les méthodes de recherche directe directionnelle (RDD). Nous décrivons en particulier les méthodes de la famille des recherches directes à treillis adaptatif, et la façon dont elles traitent les contraintes. Nous abordons également les garanties théoriques de convergence dont bénéficient ces méthodes.

2.1.1 Recherches directes directionnelles, algorithme MADS

Deux grandes approches de l'optimisation sans dérivées existent (et bien souvent, se complètent). La première cherche à reconstituer les objets précieux dont la DFO est privée : les gradients (et parfois les hessiennes) de l'objectif et des contraintes. Ces méthodes sont dites *à modèles*, et regroupent des approches déterministes et bayésiennes. La seconde approche concentre ses efforts sur une exploration de l'espace des variables guidée uniquement par les valeurs réellement observées de l'objectif et des contraintes. Ces méthodes sont dites de

recherche directe [44]. Les frontières entre ces approches sont poreuses, et de nombreuses méthodes font usage des deux simultanément, mais il est utile de poser ce constat pour débiter. Les méthodes qui nous intéressent ici sont celles de la seconde famille. Parmi elles, les recherches directes *directionnelles* (RDD) se basent sur une discrétisation de l'espace des variables.

Recherche par coordonnées. L'histoire des méthodes de RDD débute avec la *recherche par coordonnées* (*coordinate search*, abrégée Cs ci-après) [40], capable de résoudre un problème d'optimisation non contraint en se déplaçant itérativement selon les directions de la base canonique de $\mathbb{R}^n$. Le fonctionnement de cette méthode se résume comme suit : en partant d'un point $x^0 \in \mathbb{R}^n$ et d'un pas $\delta^0 \in \mathbb{R}_+^*$, évaluer la fonction f en les points $\{x^0 \pm \delta^0 e_i : i \in \llbracket 1, n \rrbracket\}$ où e_i est le $i^{\text{ième}}$ vecteur de la base canonique de $\mathbb{R}^n$. Si un point t parmi ceux-ci vérifie $f(t) < f(x^0)$, se déplacer en $x^1 = t$ et recommencer l'opération à partir de x^1 avec le pas $\delta^1 = \delta^0$. Sinon, rester en $x^1 = x^0$ et recommencer l'opération plus près de ce point en fixant $\delta^1 = \frac{1}{2}\delta^0$. Cette logique est répétée jusqu'à ce qu'une solution satisfaisante soit obtenue.

La formulation de Cs fait déjà ressortir la dichotomie fondamentale pour les méthodes de RDD : en partant d'un *itéré courant* x^k ($k \in \mathbb{N}$), elles doivent pouvoir se déplacer rapidement vers des points potentiellement éloignés de x^k faisant décroître la valeur de f . On parle de capacité d'*exploration*. Mais d'autre part, elles doivent pouvoir *raffiner* l'information locale : si une méthode est incapable d'identifier un point $t \in \mathbb{R}^n$ tel que $f(t) < f(x^k)$ à une certaine itération $k \in \mathbb{N}$, alors elle doit investiguer l'optimalité de x^k en s'intéressant à de plus petits déplacements, d'où $\delta^{k+1} = \frac{1}{2}\delta^k$. On parle de capacité d'*exploitation*. Ce constat fait naître le successeur de Cs.

Recherche par motif. La *recherche par motif* (*Generalized Pattern Search*, GPS) corrige deux des limitations de Cs : premièrement, elle autorise le pas δ^k de la recherche à augmenter en cas de succès, favorisant un déplacement plus rapide dans des zones de $\mathbb{R}^n$ prometteuses. Deuxièmement, elle étend les directions dans lesquelles f est évaluée autour de x^k : au lieu des directions $\pm e_i$, GPS considère une liste fixe $\mathbb{D} \subset \mathbb{R}^n$ dans laquelle elle choisit à chaque itération $k \in \mathbb{N}$ un sous-ensemble $\mathbb{D}^k \subseteq \mathbb{D}$. Les points évalués autour de x^k sont alors ceux de l'ensemble $P^k \triangleq \{x^k + \delta^k d : d \in \mathbb{D}^k\}$. Il est cependant nécessaire de s'assurer que les directions nécessaires couvrent suffisamment bien l'espace étudié. Pour ce faire, la définition suivante est utile.

Définition 2.1 (Ensemble engendrant positivement l'espace [15, Définition 6.1]). *Soit $\mathbb{D} = \{d_1, d_2, \dots, d_p\} \subset \mathbb{R}^n$. On dit que $\mathbb{D}$ engendre positivement $\mathbb{R}^n$ si*

$$\left\{ \sum_{i=1}^p \lambda_i d_i : \lambda_i \in \mathbb{R}_+, \forall i \in \llbracket 1, p \rrbracket \right\} = \mathbb{R}^n.$$

GPS introduit une autre idée fondamentale : s'il est possible d'identifier rapidement un point $t \notin P^k$ tel que $f(t) < f(x^k)$, il faut pouvoir en tirer profit. Chaque itération de l'algorithme est donc divisée en deux étapes.

1. L'étape de *recherche* (*search*) autorise toute opération capable d'identifier un point améliorant la valeur de l'objectif.
2. Si cette première étape échoue, l'étape de *sonde* (*poll*) évalue les points de P^k . P^k est donc appelé l'*ensemble de sonde*.

L'inconvénient de l'étape de recherche est qu'elle fournit un contrôle bien moindre sur les points évalués, contrairement à l'étape de sonde qui les contraint par les directions $\mathbb{D}^k$ et le pas δ^k . GPS intègre ces deux opérations dans un même algorithme en imposant que tout point identifié par une étape de recherche ou de sonde soit situé sur une même discrétisation de $\mathbb{R}^n$, appelée *treillis*.

Définition 2.2 (Treillis [15, Définition 8.1]). *Soient $G \in \mathbb{R}^{n \times n}$ une matrice inversible et $Z = \begin{bmatrix} z_1 & z_2 & \dots & z_p \end{bmatrix} \in \mathbb{Z}^{n \times p}$ une matrice dont les colonnes engendrent positivement $\mathbb{R}^n$. Si $D = GZ \in \mathbb{R}^{n \times p}$, on appelle treillis généré par D , centré en le point $x^k \in \mathbb{R}^n$ et de granularité $\delta^k > 0$, l'ensemble*

$$M^k \triangleq \left\{ x^k + \delta^k D y : y \in \mathbb{N}^p \right\}.$$

δ^k est aussi appelé le paramètre de taille du treillis M^k .

Cette définition se prête tout à fait à la structure de GPS : si les colonnes de la matrice D sont les directions de $\mathbb{D}^k \subseteq \mathbb{D}$ retenues à l'itération $k \in \mathbb{N}$, l'ensemble de sonde vérifie naturellement $P^k \subset M^k$. GPS impose que tout point $t \in \mathbb{R}^n$ identifié par l'étape de recherche vérifie également $t \in M^k$.

Si GPS est une amélioration considérable de CS, l'ensemble des directions qu'il est capable d'explorer reste toujours restreint. Cette limitation peut être résolue en découplant légèrement les phases de recherche et de sonde.

Recherche directe sur treillis adaptatif. Le contrôle qu'exerce un algorithme de RDD sur les points qu'il évalue est étroitement lié aux valeurs du paramètre δ^k de taille du treillis. Dans GPS, ce paramètre contrôle non seulement le treillis, mais limite la taille des pas effectués autour de x^k pour construire l'ensemble de sonde $P_{\text{GPS}}^k = \{x^k + \delta^k d : d \in \mathbb{D}^k\}$. L'idée de découpler la granularité du treillis des pas autorisés pendant la phase de sonde est à l'origine des méthodes de recherche directe sur treillis adaptatif (*Mesh Adaptive Direct Search*, MADS) [12]. Un second paramètre est donc introduit.

Définition 2.3 (Cadre [15, Définition 8.2]). *Soit M^k le treillis généré par une matrice $D = [d_1 \ d_2 \ \dots \ d_p] \in \mathbb{R}^{n \times p}$, centré en $x^k \in \mathbb{R}^n$ et de paramètre de taille $\delta^k > 0$. Si $\Delta^k \geq \delta^k$, on appelle cadre de taille Δ^k généré par D et centré en x^k l'ensemble*

$$F^k \triangleq \{x \in M^k : \|x - x^k\|_\infty \leq \Delta^k b\}$$

avec $b \triangleq \max\{\|d_i\| : i \in \llbracket 1, p \rrbracket\}$.

Δ^k est appelé le paramètre de taille du cadre F^k .

Dans MADS, le cadre est utilisé comme outil de contrôle de l'ensemble de sonde. En préservant l'inégalité $\Delta^k \geq \delta^k$ pour tout $k \in \mathbb{N}$, le paramètre de taille de cadre décroît plus lentement que la granularité du treillis. Ceci permet d'explorer un ensemble de directions de sonde plus riche. La structure de MADS est présentée formellement à l'algorithme 2.1, dans le cas d'un problème non contraint.

MADS est un cadre général, dans lequel de nombreuses stratégies de recherche et de sonde peuvent être développées. Parmi ces stratégies, citons celle dite *spéculative* : lorsqu'une étape de recherche réussit à identifier un point t tel que $f(t) < f(x^k)$, l'ensemble de recherche à l'itération suivante est le singleton

$$S^{k+1} = \{t + \alpha(t - x^k)\}$$

où $\alpha \in \{2, 4\}$. Cette stratégie est observée dans les expériences numériques menées au chapitre 5. Précisons que MADS est en constante évolution. Sa version la plus récente est celle présentée dans [18].

Nous abordons maintenant les garanties théoriques de convergence qu'offre MADS.

Algorithme 2.1 : Recherche directe sur treillis adaptatif (MADS) [15, Algorithme 8.1]

Data : $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, point initial $x^0 \in \Omega$.

- 1 **0. Initialisation**
 - 2 $\Delta^0 \in]0, \infty[$ Paramètre de taille de cadre
 - 3 $\mathbb{D} = \{d_1, d_2, \dots, d_p\} \subset \mathbb{R}^n$ engendrant positivement $\mathbb{R}^n$
 - 4 $D = \begin{bmatrix} d_1 & d_2 & \dots & d_p \end{bmatrix}$ Matrice générant le treillis
 - 5 $\tau \in \mathbb{Q} \cap]0, 1[$
 - 6 $\varepsilon_{\text{stop}} \in [0, \infty[$ critère d'arrêt
 - 7 $k \leftarrow 0$ compteur
- 8 **1. Mise à jour du paramètre**
 - 9 $\delta^k \leftarrow \min \left\{ \Delta^k, \frac{(\Delta^k)^2}{\Delta^0} \right\}$
- 10 **2. Recherche**
 - 11 **si** on trouve $t \in S^k$ (un sous-ensemble fini quelconque de M^k) tel que
 $f(t) < f(x^k)$ **alors**
 - 12 $x^{k+1} \leftarrow t$ et $\Delta^{k+1} \leftarrow \tau^{-1} \Delta^k$
 - 13 Aller en 4.
 - 14 **sinon**
 - 15 Aller en 3.
- 16 **3. Sonde**
 - 17 Sélectionner un ensemble $\mathbb{D}_\Delta^k \subseteq \mathbb{D}$ engendrant positivement $\mathbb{R}^n$ et tel que
 $P^k \triangleq \{x^k + \delta^k d, d \in \mathbb{D}_\Delta^k\}$ soit un sous-ensemble du cadre F^k .
 - 18 **si** on trouve $t \in P^k$ tel que $f(t) < f(x^k)$ **alors**
 - 19 $x^{k+1} \leftarrow t$ et $\Delta^{k+1} \leftarrow \tau^{-1} \Delta^k$
 - 20 **sinon**
 - 21 $x^{k+1} \leftarrow x^k$ et $\Delta^{k+1} \leftarrow \tau \Delta^k$
 - 22
 - 23 Aller en 4.
- 24 **4. Arrêt**
 - 25 **si** $\Delta^{k+1} \geq \varepsilon_{\text{stop}}$ **alors**
 - 26 $k \leftarrow k + 1$
 - 27 Aller en 1.
 - 28 **sinon**
 - 29 Fin.

2.1.2 Analyse de convergence de MADS

Dans le cas où le critère d'arrêt de l'algorithme 2.1 est désactivé, l'analyse de convergence de MADS s'intéresse au comportement asymptotique de la méthode. Sous l'hypothèse que

les ensembles de sous-niveau de la fonction objectif f sont bornés, plusieurs résultats sont obtenus. Le premier concerne la suite des paramètres de taille du treillis.

Proposition 2.1 ([12, Proposition 3.1]). *Soient $\{\delta^k\}_{k \in \mathbb{N}}$ et $\{\Delta^k\}_{k \in \mathbb{N}}$ les suites de paramètres de taille de treillis et de cadre produits par l'application d'une instance de MADS à une fonction f telle que l'ensemble $\{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ est borné. On a alors*

$$\liminf_{k \rightarrow \infty} \delta^k = \liminf_{k \rightarrow \infty} \Delta^k = 0.$$

Cette proposition indique qu'il existe un sous-ensemble infini d'indices $K \subseteq \mathbb{N}$ tel que $\lim_{k \in K} \delta^k = 0$. Parmi les itérés $\{x^k\}_{k \in K}$ associés à ces indices, il en existe donc une infinité pour lesquels les étapes de recherche et de sonde ont échoué à améliorer la valeur de f . De tels itérés sont appelés des *optima locaux du treillis*, et permettent d'introduire une notion fondamentale.

Définition 2.4 (Suite raffinante, point raffinant [15, Définition 7.3]). *Soient $K \subseteq \mathbb{N}$ et $\{x^k\}_{k \in K}$ une sous-suite convergente d'optima locaux du treillis. $\{x^k\}_{k \in K}$ est appelée une sous-suite raffinante si et seulement si $\lim_{k \in K} \delta^k = 0$. Le point $\hat{x} \triangleq \lim_{k \in K} x^k$ est alors appelé le point raffiné associé.*

Dans la lignée de cette définition, nous introduisons la notion de *direction raffinante*.

Définition 2.5 (Direction raffinante [15, Définition 8.4]). *Soient $K \subseteq \mathbb{N}$ et $\{x^k\}_{k \in K}$ une sous-suite raffinante. Une direction $d \in \mathbb{R}^n$ est appelée direction raffinante s'il existe un ensemble infini d'indices $L \subseteq K$ tel que les directions de sondes $d^\ell \in \mathbb{D}_\Delta^\ell$ explorées par MADS vérifient*

$$\lim_{\ell \in L} \frac{d^\ell}{\|d^\ell\|} = \frac{d}{\|d\|}.$$

Une direction raffinante peut s'interpréter comme une direction que MADS a explorée un nombre infini de fois durant sa convergence vers un point raffiné, et qui a systématiquement échoué à améliorer la valeur de f . Ceci incite à penser que

$$f'(\hat{x}; d) \triangleq \lim_{t \searrow 0} \frac{f(\hat{x} + td) - f(\hat{x})}{t} \geq 0.$$

Cependant, aucune hypothèse n'est faite sur la différentiabilité de f ; il n'y a donc aucune garantie sur l'existence de cette dérivée directionnelle. C'est pourquoi l'optimisation sans

dérivées a recours à une dérivée directionnelle *généralisée*, issue du domaine de l'analyse non lisse.

Définition 2.6 (Dérivée directionnelle généralisée de Clarke [27]). *Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction lipschitzienne au voisinage d'un point $x \in \mathbb{R}^n$. Soit $d \in \mathbb{R}^n$. La dérivée directionnelle généralisée de Clarke de f en x dans la direction d est la quantité*

$$f^\circ(x; d) \triangleq \limsup_{\substack{y \rightarrow x \\ t \searrow 0}} \frac{f(y + td) - f(y)}{t}.$$

Le théorème suivant confirme à l'aide de cet outil ce que l'intuition nous dictait plus tôt.

Théorème 2.2 (Convergence de MADS [12, Théorème 3.12]). *Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $\hat{x} \in \mathbb{R}^n$ un point raffiné obtenu par l'application de MADS à la minimisation de f . Soit $d \in \mathbb{R}^n$ une direction raffinante associée à $\hat{x}$. Si f est lipschitzienne au voisinage de $\hat{x}$, alors*

$$f^\circ(\hat{x}; d) \geq 0.$$

La présentation donnée ici des méthodes du type de MADS aborde la résolution de problèmes non contraints. Nous évoquons maintenant les capacités de ces méthodes à traiter des problèmes en présence de contraintes.

2.1.3 Traitement des problèmes contraints avec MADS

Une première stratégie pour traiter le problème contraint

$$\min_{x \in \Omega} f(x) \tag{2.1}$$

avec $\Omega \subseteq \mathbb{R}^n$ consiste à exécuter MADS en remplaçant la fonction objectif f par la fonction suivante, appelée fonction de *barrière extrême* :

$$\forall x \in \mathbb{R}^n, \quad f_\Omega(x) \triangleq \begin{cases} f(x) & \text{si } x \in \Omega, \\ \infty & \text{sinon.} \end{cases}$$

Cette approche fournit également des garanties théoriques de convergence, exprimées grâce à un autre outil issu de l'analyse non lisse : le cône hypertangent au domaine Ω . Rappelons que la notation $B(x, \varepsilon) \triangleq \{y \in \mathbb{R}^n : \|x - y\| < \varepsilon\}$ désigne la boule ouverte de $\mathbb{R}^n$ centrée en $x \in \mathbb{R}^n$ et de rayon $\varepsilon > 0$.

Définition 2.7 (Cône hypertangent [15, Définition 6.7]). Soient $\Omega \subseteq \mathbb{R}^n$ et $x \in \Omega$. Un vecteur $d \in \mathbb{R}^n$ est dit hypertangent à Ω en x s'il existe $\varepsilon > 0$ tel que

$$\forall y \in \Omega \cap B(x, \varepsilon), \forall w \in B(d, \varepsilon), \forall t \in]0, \varepsilon[, \quad y + tw \in \Omega.$$

L'ensemble des vecteurs hypertangents à l'ensemble Ω en x est un cône, appelé le cône hypertangent à Ω en x , et noté $T_\Omega^H(x)$.

À partir de cette définition, tous les concepts décrivant la convergence de MADS se généralisent au cas contraint. Avec les mêmes notations qu'à la définition 2.5, une direction $d \in \mathbb{R}^n$ est dite raffinante si $x^\ell + \delta^\ell d^\ell \in \Omega$ pour tout $\ell \in L$ et $\lim_{\ell \in L} d^\ell / \|d^\ell\| = d / \|d\|$. L'application de MADS au problème (2.1) avec la fonction de barrière extrême produit un résultat de convergence très similaire au cas non contraint.

Théorème 2.3 (Convergence de MADS dans le cas contraint). Soit $\hat{x} \in \Omega$ un point raffiné obtenu par l'application de MADS à la minimisation de f_Ω . Soit $d \in T_\Omega^H(\hat{x})$ une direction raffinante associée à $\hat{x}$. Si f est lipschitzienne près de $\hat{x}$, alors

$$f^\circ(\hat{x}; d) \geq 0.$$

Ces résultats de convergence sont utilisés pour démontrer certaines propriétés de la méthode développée au chapitre 4.

La méthode de la barrière extrême n'est pas la seule permettant à MADS de résoudre le problème (2.1). La méthode dite de la *barrière progressive* [13] suppose qu'il existe $X \subseteq \mathbb{R}^n$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ($m \in \mathbb{N}$) tels que $\Omega = \{x \in X : g_i(x) \leq 0, i \in \llbracket 1, m \rrbracket\}$, et utilise la fonction suivante, appelée fonction de violation de contraintes :

$$\forall x \in \mathbb{R}^n, \quad c(x) \triangleq \begin{cases} \sum_{i=1}^m (\max\{g_i(x), 0\})^2 & \text{si } x \in X \\ \infty & \text{sinon.} \end{cases}$$

À l'itération $k \in \mathbb{N}$, l'infaisabilité des points évalués est mesurée grâce à c , et trois issues sont possibles :

1. si un point évalué $t \in M^k$ est tel que $c(t) < c(x^k)$ et $f(t) < f(x^k)$, l'itération est dite *dominante* et $\Delta^{k+1} \geq \Delta^k$;
2. si un point évalué $t \in M^k$ est tel que $c(t) < c(x^k)$ et $f(t) \geq f(x^k)$, l'itération est dite *améliorante* et $\Delta^{k+1} = \Delta^k$;

3. si aucun de ces deux cas ne se produit, l'itération est un *échec*, et $\Delta^{k+1} < \Delta^k$.

Les implémentations gardent généralement le compte des meilleures valeurs de f générées par des points réalisables et non réalisables, et des stratégies supplémentaires peuvent venir raffiner le traitement des contraintes.

La notion de sous-suite raffinante peut être étendue au cadre de la barrière progressive. Si $\hat{x}$ est un point raffiné, des garanties théoriques de convergence peuvent être énoncées sur les directions de $T_X^H(\hat{x})$ et de $T_\Omega^H(\hat{x})$. Nous ne les mentionnons pas ici.

Notons également qu'une approche hybride nommée barrière *progressive à extrême* existe [14]. Dans cette approche, les violations de contraintes du point de départ sont traitées par la barrière progressive. Dès lors qu'une contrainte est satisfaite, son traitement est délégué à la barrière extrême.

Les méthodes de la barrière extrême et de la barrière progressive peuvent être particulièrement efficaces lorsque Ω est défini par des contraintes d'inégalité. La section suivante montre que ce n'est pas le cas lorsque des égalités interviennent dans la définition de Ω .

2.1.4 Limitations en présence d'égalités

Une approche classique pour résoudre un problème d'optimisation du type de (P) consiste à le reformuler en

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.c.} \quad & g(x) \leq 0 \\ & h(x) \leq 0 \\ & -h(x) \leq 0. \end{aligned} \tag{P_{\leq}}$$

Cette formulation peut être traitée avec MADS, par les méthodes de la barrière extrême ou progressive. Le théorème 2.3 est toujours vrai ; il souffre simplement du constat suivant : dans la plupart des cas, le cône hypertangent au domaine réalisable du problème $(P_{\leq})$ est vide en chacun de ses points. La garantie de convergence porte donc sur des directions qui n'existent pas. Ce fait est connu dans le cas où h est une fonction linéaire [16], et des exemples simples existent dans le cas non linéaire. Considérons un problème dont la solution doit être de norme unitaire :

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & x_1 \\ \text{s.c.} \quad & x_1^2 + x_2^2 = 1. \end{aligned}$$

La solution globale de ce problème est $x^* = (-1, 0)$, et son domaine réalisable Ω est le cercle unité. Montrons que $T_\Omega^H(x^*) = \emptyset$. Avec les notations de la définition 2.7, soient $\varepsilon > 0$, $d \in \mathbb{R}^n$ et $w \in B(d, \varepsilon)$. On peut considérer le cas où $y = x \in B(x, \varepsilon)$. $x + tw \in \Omega$ si et seulement si

le déplacement de x à $x + tw$ est une corde du cercle unité. Il en résulte que $x + \frac{t}{2}w \notin \Omega$: il est impossible que le point milieu de cette corde appartienne également au cercle. Ceci étant vrai quel que soit $\varepsilon > 0$, aucune direction $d \in \mathbb{R}^n$ ne peut satisfaire la définition 2.7 : le cône hypertangent en x^* est vide. Ce raisonnement peut en réalité être répété pour tout $x \in \Omega$.

Cette limitation se vérifie en pratique : la littérature a expérimenté les limites de l'algorithme MADS sur les reformulations du type de $(P_{\leq})$ [20].

En présence d'égalités, les limites d'un algorithme dont les garanties de convergence reposent sur la notion de cône hypertangent s'illustrent donc à la fois en théorie et en pratique. C'est ce qui motive l'approche géométrique de l'adaptation de MADS proposée au chapitre 4.

Cette première section s'est consacrée exclusivement aux recherches directes directionnelles (et plus spécifiquement à MADS) pour en montrer les limitations en présence d'égalités. Ce qui est présenté ici est loin d'être exhaustif : de très nombreuses variantes des méthodes de recherche directe existent, dont certaines sont dédiées à la résolution de problèmes spécifiques. Le lecteur intéressé trouvera un l'état de l'art plus complet dans la référence [38]. Nous évoquons maintenant le traitement des contraintes d'égalité dans le reste de la littérature, en commençant par les approches non réalisables.

2.2 Méthodes non réalisables

Que ce soit dans la littérature de l'optimisation avec ou sans dérivées, les méthodes numériques le plus utilisées à ce jour pour résoudre des problèmes de la forme de (P) dans le cas où g et/ou h sont des fonctions non linéaires, partagent une démarche commune : elles peuvent *relâcher* ces contraintes. Par-là, on veut dire que les suites d'itérés qu'elles génèrent peuvent contenir – et généralement, contiennent – des points non réalisables pour (P).

Chaque itéré, réalisable ou non, fournit une information locale que l'algorithme utilise pour guider l'optimisation. Dans le cas de l'optimisation avec dérivées, cette information locale implique le plus souvent les dérivées premières voire secondes de f , g et h .

Malgré l'omniprésence des dérivées dans les aspects théoriques et pratiques de l'optimisation sous contraintes, la littérature propose des homologues sans dérivées de plusieurs méthodes classiques. De telles méthodes supposent inaccessibles la forme et les gradients (à plus forte raison, les hessiennes) de f , g et h , mais certaines parviennent tout de même à offrir des résultats théoriques de convergence.

Les premières méthodes que nous abordons sont les méthodes de région de confiance. Leur description nécessite l'introduction des notions de *modèles* linéaires et quadratiques d'une fonction.

2.2.1 Interpolation, gradients et hessiennes simples

Gradient et hessienne d'une fonction capturent sa pente et sa courbure locales, ce sont donc des ingrédients précieux pour retraduire son comportement autour d'un point. À cette fin, la branche de l'optimisation sans dérivées travaillant sur les modèles cherche à en obtenir des estimations par *interpolation*. Nous présentons les principes de l'interpolation linéaire et de l'interpolation quadratique en suivant les notations de [15, Chapitre 9].

Si $f : \mathbb{R}^n \rightarrow \mathbb{R}$, une fonction linéaire approximant f est donnée pour tout $x \in \mathbb{R}^n$ par

$$L(x) \triangleq \alpha_0 + \alpha^\top x, \quad (2.2)$$

où $\alpha_0 \in \mathbb{R}$ et $\alpha \in \mathbb{R}^n$. Notons $\mathbb{Y} \triangleq \{y_0, y_1, \dots, y_p\} \subset \mathbb{R}^n$ un ensemble de $p + 1$ points d'interpolation, avec $p \geq n$. Les paramètres (α_0, α) sont alors déterminés en résolvant le système linéaire

$$\begin{bmatrix} 1 & y_0^\top \\ 1 & y_1^\top \\ \vdots & \vdots \\ 1 & y_p^\top \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha \end{bmatrix} = \begin{bmatrix} f(y_0) \\ f(y_1) \\ \vdots \\ f(y_p) \end{bmatrix}.$$

Si la famille $\{y_1 - y_0, y_2 - y_0, \dots, y_p - y_0\}$ est libre, ce système admet une solution unique (potentiellement au sens des moindres carrés). L'ensemble $\mathbb{Y}$ est alors dit *propice pour l'interpolation linéaire*. L est appelée un *modèle linéaire* de f en y_0 et le vecteur α est appelé le *gradient simplexe* de f en y_0 dans les directions $\{y_1 - y_0, y_2 - y_0, \dots, y_p - y_0\}$.

Une démarche semblable permet d'obtenir un *modèle quadratique* de f sous la forme

$$Q(x) = \alpha_0 + \alpha^\top x + \frac{1}{2} x^\top H x,$$

Nous détaillons plusieurs familles de méthodes dans lesquelles les modèles sont utilisés.

2.2.2 Modèles linéaires : COBYLA

Les méthodes de région de confiance sont d'un usage très courant en optimisation avec gradients [29]. Elles approximent les fonctions définissant un problème d'optimisation par des modèles locaux, et estiment la région dans laquelle ces modèles coïncident avec les valeurs observées lors des évaluations. Plusieurs méthodes de région de confiance sans dérivées ont été conçues par Michael J.D. Powell, et sont aujourd'hui connues sous le nom de *méthodes de Powell*.

La cadette de ces méthodes est connue sous le nom de COBYLA, pour *Constrained Optimization BY Linear Approximation* [60]. COBYLA ne fait usage que de modèles linéaires de l'objectif et des contraintes, et met l'accent sur la géométrie des ensembles utilisés pour l'interpolation. L'interpolation étant linéaire, ces ensembles doivent former des simplexes de $\mathbb{R}^n$. COBYLA évalue la qualité d'un point par une fonction dite *de pénalité*¹, et chaque itéré est généré dans une région de confiance centrée en le meilleur sommet du simplexe d'interpolation courant, au sens de la fonction de mérite. Une itération $k \in \mathbb{N}$ de COBYLA requiert :

- ▷ un simplexe $\mathbb{Y}^k \subset \mathbb{R}^n$ non dégénéré (i.e. $n + 1$ sommets linéairement indépendants) ;
- ▷ un rayon de confiance $\rho^k > 0$;
- ▷ un paramètre de pénalité $\mu^k > 0$ déterminant la fonction de mérite $\Phi^k : \mathbb{R}^n \rightarrow \mathbb{R}$.

On note $x_*^k \in \arg \min_{x \in \mathbb{Y}^k} \Phi^k(x)$ un meilleur point de $\mathbb{Y}^k$ au sens de Φ . L'itération suit alors le schéma suivant :

1. À partir de $\mathbb{Y}^k$, calculer $(\tilde{f}^k, \tilde{g}^k, \tilde{h}^k)$, des modèles linéaires pour (f, g, h) .
2. Résoudre le problème de région de confiance

$$\begin{aligned} x^k \in \arg \min_{x \in \mathbb{R}^n} \quad & \tilde{f}^k(x) \\ \text{s.c.} \quad & \tilde{g}^k(x) \leq 0 \\ & \tilde{h}^k(x) \leq 0 \\ & -\tilde{h}^k(x) \leq 0 \\ & \|x - x_*^{k-1}\| \leq \rho^k \end{aligned} \quad . \quad (2.3)$$

3. Évaluer les fonctions de pénalité réelle et estimée :

$$\Phi^k(x^k) = f(x^k) + \mu^k \max \left\{ \|[g(x^k)]_+\|_\infty, \|h(x^k)\|_\infty \right\}$$

$$\tilde{\Phi}^k(x^k) = \tilde{f}(x^k) + \mu^k \max \left\{ \|[\tilde{g}(x^k)]_+\|_\infty, \|\tilde{h}(x^k)\|_\infty \right\}$$

où $[x]_+ = \max\{0, x\}$.

4. Calculer l'amélioration relative

$$\gamma \triangleq \frac{\Phi^k(x^k) - \Phi^k(x_*^k)}{\tilde{\Phi}^k(x^k) - \tilde{\Phi}^k(x_*^k)}.$$

1. La notion de *fonction de pénalité* sera définie plus loin, ayant donné lieu à une famille entière de méthodes. La fonction de mérite utilisée par COBYLA ne joue pas un rôle central dans son développement théorique.

5. Selon la valeur de γ , mettre à jour μ^k , ρ^k et $\mathbb{Y}^k$. Si besoin, appliquer une procédure pour améliorer la géométrie de $\mathbb{Y}^{k+1}$. Ordonner ses sommets.

Diverses stratégies de mise à jour de ρ^k et μ^k , ainsi que les procédures utiles à l'amélioration de la géométrie de $\mathbb{Y}^k$, sont empruntées aux idées développées dans l'« autre méthode du simplexe » de Nelder et Mead [55].

À notre connaissance, la littérature ne recèle aucun résultat théorique sur la convergence de COBYLA. Powell lui-même indique en 2007 qu'il est « incapable de donner une réponse positive à cette question en les termes usuels de la théorie de la convergence » [61]. COBYLA reste cependant très utilisée. Elle est implémentée aux côtés de quatre autres méthodes de Powell dans la librairie PRIMA [70].

2.2.3 Modèles quadratiques : SQP

L'approche par *programmation quadratique successive* constitue une famille d'approches connues pour la résolution de problèmes non linéaires avec gradients. Cette approche est connue dans la littérature sous l'abréviation SQP pour *Sequential Quadratic Programming*, que nous réutiliserons ici. Ces méthodes résolvent successivement des sous-problèmes basés sur des approximations quadratiques du lagrangien de (P), sans région de confiance [56].

Des versions sans dérivées de méthodes SQP ont été proposées, avec des sous-problèmes construits à partir de modèles quadratiques de l'objectif et des contraintes [66, 67, 63]. Les sous-problèmes sont résolus dans des régions de confiance, et habituellement en deux étapes selon l'approche de Byrd-Omojokun [58] : une étape favorisant la faisabilité, et une autre favorisant l'amélioration de l'objectif. Ces méthodes ne sont pas munies de garanties de convergence.

Une autre méthode SQP pour l'optimisation sans dérivées fut proposée par l'un des descendants académiques de Powell. En reprenant le schéma de nomination de ses méthodes, Ragonneau [62] nomme cette méthode COBYQA, pour *Constrained Optimization BY Quadratic Approximation*. Elle se différencie de COBYLA par l'utilisation de modèles quadratiques, mais aussi par sa façon de construire et de mettre à jour l'ensemble d'interpolation $\mathbb{Y}^k$. COBYQA propose également un traitement explicite des bornes.

Enfin, [54] étudie spécifiquement l'utilisation d'une méthode SQP pour le traitement de problèmes stochastiques en présence de contraintes d'égalité.

2.2.4 Méthodes de pénalité et de barrière

On appelle *méthode de pénalité* une méthode qui fait du problème (P) un problème non contraint, en ajoutant à sa fonction objectif des quantités dépendant des valeurs des contraintes. Nous décrivons deux versions sans dérivées de ces méthodes.

Lagrangien augmenté. La méthode du lagrangien augmenté a été créée pour pallier le mauvais conditionnement de la pénalité quadratique donnée, pour $\mu > 0$ et $x \in \mathbb{R}^n$, par

$$Q(x; \mu) \triangleq f(x) + \sum_{i=1}^p h_i(x)^2 + \frac{\mu}{2} \sum_{j=1}^m (\max\{0, g_j(x)\})^2,$$

en y ajoutant des estimés des multiplicateurs de Lagrange optimaux [56].

Plusieurs variantes sans dérivées de la méthode du lagrangien augmenté ont été proposées [47, 48, 33, 42]. Nous citons en particulier la fonction de pénalité utilisée dans [33, 48]. Cette fonction est définie pour tous $\rho \in \mathbb{R}_+^*$, $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^p$ et $\mu \in \mathbb{R}_+^m$, par :

$$L(\rho; x, \lambda, \mu) \triangleq f(x) + \frac{\rho}{2} \left(\sum_{i=1}^p \left(h_i(x) + \frac{\mu_i}{\rho} \right)^2 + \sum_{j=1}^m \max \left\{ 0, g_j(x) + \frac{\lambda_j}{\rho} \right\}^2 \right).$$

Une méthode est alors proposée, avec des formules pour l'estimation des multiplicateurs ainsi que des règles de mise à jour du paramètre de pénalité $\rho > 0$. Les auteurs énoncent des hypothèses sous lesquelles la suite des paramètres de pénalité est bornée, et les solutions obtenues satisfont les conditions de Karush-Kuhn-Tucker (KKT) du problème (P). Parmi ces conditions, il est nécessaire que $f, g, h \in \mathcal{C}^1$.

Méthodes de barrière. Les méthodes de barrière sont les précurseurs de ce qui est connu aujourd'hui sous le nom de méthodes de *points intérieurs*. Une méthode de barrière sans dérivées est proposée dans [49], puis retravaillée dans [24] sous forme d'une barrière *logarithmique*. La fonction de pénalité utilisée combine deux approches : elle traite les contraintes d'inégalité par une barrière dite logarithmique, et les contraintes d'égalité par une pénalité externe. Le résultat est de la forme suivante, pour $x \in \mathbb{R}^n$ et $\varepsilon \in \mathbb{R}_+^*$ et $\nu \in]1, 2]$:

$$P(x, \varepsilon) \triangleq f(x) - \varepsilon \sum_{j=1}^m \log(-g_j(x)) + \frac{1}{\varepsilon} \sum_{i=1}^p |h_i(x)|^\nu.$$

Sous certaines hypothèses, dont $f, g, h \in \mathcal{C}^1$, les auteurs prouvent que tout point d'accumulation de la suite des itérés générés par la méthode proposée satisfait les conditions de KKT

du problème (P).

Ce dernier travail fait l'objet d'une implémentation récente dans laquelle les problèmes de minimisation de l'objectif avec barrière logarithmique sont résolus avec MADS [11]. La version qui y est proposée diffère légèrement de la version initiale de [24] : les contraintes d'inégalité y sont séparées en deux ensembles, et certaines sont traitées par la pénalité externe. La méthode obtenue est dénommée MADS-PIP, pour *Penalty-Interior Point*.

2.2.5 Méthodes de restauration inexacte

Une famille de méthodes se situe à la frontière entre méthodes réalisables et non réalisables : les méthodes de restauration inexacte. Celles-ci décomposent chaque itération en deux phases : la phase dite de *restauration* cherche à identifier un point $y^k \in \mathbb{R}^n$ réalisable sans se soucier de la fonction objectif, tandis que la phase de minimisation résout une relaxation du problème (P) en partant de y^k , dans l'espoir d'améliorer la valeur de l'objectif en préservant le caractère réalisable du point obtenu [25, 41]. Ces méthodes supposent que les contraintes sont connues, contrairement à l'objectif. Elles sont donc particulièrement pertinentes dans le cadre où l'évaluation de l'objectif est coûteuse : elles évitent de gâcher des évaluations pour générer des points réalisables.

Dans cette approche, l'ensemble $\Omega_=_$ est parfois décrit comme un *domaine fin* (*thin domain*) [52]. Les auteurs en donnent la définition suivante : à partir d'un point $x \in \Omega_=_$, si une direction $d \in \mathbb{R}^n$ et un scalaire $\alpha > 0$ sont choisis, la probabilité que $x + \alpha d \in \Omega_=_$ est nulle. Cette idée rejoint l'exemple de limitation des recherches directes directionnelles qui est proposé dans la section 2.1.4. Les auteurs qualifient également de *fins* les domaines non connexes. Dans le cadre de l'approche par restauration inexacte, les problèmes résolus lors des phases de minimisation sont obtenus en relâchant les contraintes sous la forme $g(x) \leq \varepsilon^k$ et $|h(x)| \leq \varepsilon^k$, où $\varepsilon^k \searrow 0$. Les domaines relâchés sont alors appelés *domaines larges*, par opposition aux domaines fins.

Sous certaines hypothèses dont la continuité de l'objectif et des contraintes, les auteurs montrent que la méthode proposée est capable de générer des solutions globales du problème (P).

Nous nous intéressons maintenant aux méthodes $\Omega_=_$ -réalisables, i.e. celles qui ne cherchent à évaluer que des points réalisables au moins pour les contraintes d'égalité.

2.3 Méthodes $\Omega_{=}$ -réalisables

L'optimisation sans dérivées constitue un cadre particulier, dans lequel les valeurs des fonctions définissant le problème (P) peuvent être obtenues par des processus parfois coûteux, le premier exemple étant celui de simulations numériques.

Il est tout à fait possible que de telles simulations numériques échouent dans le cas où elles sont exécutées avec des paramètres inadéquats. Une telle situation se produit par exemple si les variables $x_1, x_2, \dots, x_n$ représentent une distribution de probabilité : les contraintes

$$\sum_{i=1}^n x_i = 1 \quad \text{et} \quad x_i \geq 0, \forall i \in \llbracket 1, n \rrbracket$$

doivent être respectées pour qu'un simulateur utilisant cette distribution puisse être exécuté. Si de telles contraintes sont connues *a priori* par l'utilisateur, il faut pouvoir ne générer que des points les respectant. Les méthodes satisfaisant cette exigence au moins pour toutes les contraintes d'égalité définissant (P) sont appelées $\Omega_{=}$ -réalisables.

2.3.1 Méthodes entièrement réalisables

Plusieurs méthodes de la littérature motivent la conception de méthodes $\Omega_{=}$ -réalisables par le constat que nous venons de dresser. Certaines s'appuient sur des régions de confiance : elles construisent des modèles linéaires ou quadratiques de f , et résolvent les sous-problèmes de région de confiance associés en imposant que toutes les contraintes du problème (P) soient strictement satisfaites par les solutions retournées [30, 28, 69].

Une autre méthode propose, dans le cas où les contraintes sont connues, de projeter les directions de recherche sur l'ensemble Ω [51]. En ceci, elles sont proches d'une implémentation de la méthode que nous développons dans le chapitre 4.

2.3.2 Cas des égalités linéaires

Une approche pour traiter les contraintes d'égalité linéaires par une méthode $\Omega_{=}$ -réalisable est proposée dans [16]. Cette approche est fondamentale pour le travail effectué dans ce mémoire.

Les contraintes considérées s'expriment sous la forme $Ax = 0$, avec $A \in \mathbb{R}^{p \times n}$. L'ensemble $\Omega_{=}$ correspond donc au noyau de la matrice A , noté $\text{Ker}(A)$: il s'agit d'un sous-espace vectoriel de $\mathbb{R}^n$ de dimension $n - \text{rang}(A)$. Les auteurs définissent pour tout $q \in \llbracket n - \text{rang}(A), n \rrbracket$ la notion de convertisseur linéaire comme une application linéaire surjective $\varphi : \mathbb{R}^q \rightarrow \text{Ker}(A)$.

Cet outil permet de reformuler (P) sous la forme

$$\begin{aligned} \min_{v \in \mathbb{R}^q} \quad & f \circ \varphi(v) \\ \text{s.c.} \quad & g \circ \varphi(v) \leq 0, \end{aligned} \tag{P}_\varphi$$

profitant ainsi d'une réduction de dimension du problème au plus égale au rang de A . Cette possibilité est très intéressante du point de vue des algorithmes de RDD, pour lesquels la dimension de l'espace de recherche peut s'avérer cruciale.

La linéarité du convertisseur φ permet d'établir une correspondance exacte entre les espaces $\mathbb{R}^q$ et $\text{Ker}(A) \subset \mathbb{R}^n$, et notamment entre les cônes hypertangents aux ensembles

$$\Omega_{\leq}^x \triangleq \{x \in \mathbb{R}^n : g(x) \leq 0\} \quad \text{et} \quad \Omega_{\leq}^v \triangleq \{v \in \mathbb{R}^q : g \circ \varphi(v) \leq 0\}.$$

Ainsi, si $v \in \Omega_{\leq}^v$, alors

$$T_{\Omega_{\leq}^x}^H(\varphi(v)) = \varphi(T_{\Omega_{\leq}^v}^H(v)).$$

En résolvant le problème $(P)_\varphi$ avec MADS, le résultat de convergence du théorème 2.3 permet d'affirmer que si $v^* \in \mathbb{R}^q$ est un point raffiné et que $x^* \triangleq \varphi(v^*)$, alors

$$\forall d \in T_{\Omega_{\leq}^x}(x^*), \quad f^\circ(x^*; d) \geq 0.$$

Ce résultat est permis par la linéarité de φ . La tâche principale de ce mémoire consiste à étendre cette technique au cas où $\Omega_{\leq}$ n'est pas un sous-espace vectoriel (ni affine) de $\mathbb{R}^n$, mais une *sous-variété riemannienne* de $\mathbb{R}^n$. Ce terme est défini au chapitre 3.

Plusieurs convertisseurs linéaires sont proposés, parmi lesquels une décomposition en valeurs singulières de A , une décomposition QR , ou encore une projection orthogonale sur $\text{Ker}(A)$. Ces convertisseurs sont utilisés au chapitre 5 pour comparer leurs performances à celles de la méthode que nous développons.

Nous achevons cette revue de littérature en mentionnant des outils classiques utilisés pour la comparaison numérique d'algorithmes sans dérivées. Ces outils seront réinvestis dans le chapitre 5, qui concerne les résultats numériques.

2.4 Comparaison numérique d'algorithmes d'optimisation

Le critère déterminant la qualité d'un algorithme d'optimisation sans dérivées n'est souvent pas sa rapidité d'exécution. Dans la plupart des cas, la fraction dominante du coût de calcul est liée à l'évaluation de l'objectif et des contraintes. C'est pourquoi les performances des al-

algorithmes de DFO sont mesurées en fonction du nombre d'évaluations nécessaires à identifier des solutions. Un outil répandu dans la communauté pour présenter ces mesures est appelé *profil de données* [53]. Puisqu'il est utilisé pour présenter les tests numériques du chapitre 5, nous l'exposons.

Supposons qu'un algorithme a est exécuté sur un problème d'optimisation p à partir d'un point de départ $x^0 \in \mathbb{R}^n$, et que la valeur optimale connue pour la fonction objectif de p soit $f^* \in \mathbb{R}$. La *précision* atteinte par l'algorithme après un nombre $N \in \mathbb{N}$ d'évaluations est le ratio

$$f_{\text{acc}}^N \triangleq \frac{f(x^N) - f(x^0)}{f^* - f(x^0)} \in [0, 1].$$

Une valeur $f_{\text{acc}}^N \approx 1$ indique que l'algorithme est proche de la valeur optimale connue. En fixant une valeur $\tau \in]0, 1[$, on dit que l'algorithme a a résolu le problème p à la tolérance τ (ou qu'il l'a τ -résolu) dès lors que $f_{\text{acc}}^N \geq 1 - \tau$. On note alors

$$T_{a,p} \triangleq \begin{cases} 1 & \text{si } a \text{ a } \tau\text{-résolu } p \text{ à partir d'un certain } N \in \mathbb{N} \\ 0 & \text{sinon.} \end{cases}$$

Si $T_{a,p} = 1$, on définit le nombre $N_{a,p} \in \mathbb{N}$ comme le plus faible nombre d'évaluations nécessaires à l'algorithme a pour τ -résoudre le problème p .

Supposons que l'on souhaite évaluer les performances d'un algorithme a sur un ensemble $\mathcal{P}$ de problèmes d'optimisation, où $n_p \in \mathbb{N}$ est la dimension du problème $p \in \mathcal{P}$. Pour ce faire, on définit la fonction suivante.

Définition 2.8 (Profil de données [53]). *À une tolérance $\tau \in]0, 1[$ donnée, on appelle profil de données de l'algorithme a sur l'ensemble de problèmes $\mathcal{P}$ la fonction $d_a : \mathbb{N} \rightarrow [0, 1]$ définie par*

$$d_a(k) \triangleq \frac{1}{|\mathcal{P}|} |\{p \in \mathcal{P} : N_{a,p} \leq k(n_p + 1)T_{a,p}\}|.$$

où $n_p \in \mathbb{N}$ est la dimension du problème $p \in \mathcal{P}$ et $|E|$ désigne le nombre d'éléments contenus dans l'ensemble E .

Le profil de données d'un algorithme illustre ses performances absolues sur l'ensemble de problèmes considéré. Pour un $k \in \mathbb{N}$, la valeur $d_a(k)$ indique la proportion des problèmes de $\mathcal{P}$ qui ont été résolus en moins de k groupes de $n_p + 1$ évaluations par l'algorithme a . Le facteur $n_p + 1$ n'est pas choisi au hasard : il s'agit du nombre d'évaluations nécessaires au calcul d'un gradient simplexe (voir la section 2.2.1).

Souvent, pour comparer les performances de plusieurs algorithmes d'un ensemble $\mathcal{A}$, leurs

profils de données sont tracés sur la même figure. Dans ce cas, la valeur optimale f^* utilisée pour calculer la précision f_{acc}^N de $a \in \mathcal{A}$ sur $p \in \mathcal{P}$ est la meilleure des valeurs identifiées par tous les algorithmes $a' \in \mathcal{A}$ ayant résolu p .

Ceci clôt la revue de littérature concernant la première grande composante de ce mémoire, à savoir l'optimisation sans dérivées sous contraintes d'égalité. Afin de décrire sans encombre la méthode proposée au chapitre 4, il est nécessaire de consacrer un chapitre à l'autre grande composante de ce travail : l'optimisation riemannienne.

CHAPITRE 3 OUTILS D’OPTIMISATION RIEMANNIENNE

Le travail effectué dans ce mémoire utilise les outils de l’*optimisation riemannienne*, qui repose sur plusieurs concepts de géométrie différentielle. Puisque ces outils sont le fondement de la méthode développée plus loin, les plus pertinents d’entre eux sont recensés ici.

Si la géométrie euclidienne est d’une utilité quotidienne indéniable, elle est limitée à l’étude des espaces non courbés – du moins, ceux qui vérifient certaines propriétés de *linéarité*. Ce sont les travaux de cartographie de C.F. Gauss qui, au XIX^{ième} siècle, ouvrent la porte à l’étude des propriétés géométriques d’ensembles courbes. La géométrie différentielle, dans ses grands traits, est l’étude des propriétés de formes *suffisamment lisses*¹, à l’image de certaines courbes et surfaces. Si elle recèle nombre d’outils complexes, seule une infime partie en est présentée ici, avec un degré de généralité adapté à l’étude de sous-ensembles de $\mathbb{R}^n$.

L’optimisation riemannienne est la discipline qui s’intéresse à l’optimisation de fonctions définies uniquement sur ces sous-ensembles *suffisamment lisses*. Puisque ces derniers sont appelés *variétés* (en anglais : *manifolds*), ils ont donné un autre nom à l’optimisation riemannienne : celui d’*optimisation sur variétés* (en anglais : *optimization on manifolds*).

Ce chapitre se veut une introduction intuitive aux outils et aux enjeux de l’optimisation riemannienne. Il présente plusieurs définitions et résultats tirés d’un ouvrage du domaine [22], plutôt que d’un ouvrage de géométrie différentielle, dont le degré d’abstraction serait inutilement élevé pour ce mémoire. Les notions présentées sont réutilisées dans la méthode développée au chapitre 4. Dans cette optique, des exemples originaux sont proposés en plus des définitions et résultats classiques. De la même manière, certains résultats simples mais centraux pour la suite de ce mémoire sont reformulés.

3.1 Sous-variétés riemanniennes de $\mathbb{R}^n$

Toute incartade dans le monde de la géométrie différentielle débute en abordant le concept de *variété*, et pour cause : il s’agit là de son objet d’étude le plus fondamental. Nous n’échappons donc pas à cette règle.

Une intuition répandue de ce qu’est une variété est la suivante : *est une variété tout ensemble qui, localement en chacun de ses points, ressemble à un espace euclidien*. Plusieurs termes restent à préciser, mais ce point de départ est excellent dans le cadre qui nous occupe. Il est à remarquer que cette définition ne fait mention d’aucun *espace ambiant* : pour qu’un

1. Terme dont la signification reste à préciser.

objet mathématique soit une variété, nul besoin qu'il soit regardé comme un sous-ensemble d'un certain espace. Une variété est habituellement définie à travers les concepts de *cartes* et d'*atlas*². Si $\mathcal{M}$ est une variété, cartes et atlas permettent la construction de bijections locales entre $\mathcal{M}$ et l'espace euclidien, sans jamais rien supposer sur la nature de $\mathcal{M}$ ou sur un ensemble dans lequel elle serait incluse. Cette généralité s'avère peu utile dans le cadre de ce mémoire, puisque les variétés auxquelles il s'intéresse sont des ensembles réalisables de problèmes d'optimisation définis sur $\mathbb{R}^n$. Pour cette raison, les objets auxquels nous nous intéressons sont ceux appelés *sous-variétés* de $\mathbb{R}^n$. Leur présentation proposée ici suit celle de [22].

Pour qualifier ces sous-variétés, le problème est le suivant : quels sous-ensembles de $\mathbb{R}^n$ peuvent être « correctement » linéarisés en chacun de leurs points ? L'exemple canonique d'un tel ensemble est celui de la sphère unité de $\mathbb{R}^n$, définie par

$$\mathbb{S}^{n-1} \triangleq \{x \in \mathbb{R}^n : \|x\| = 1\}.$$

Proche de $x \in \mathbb{S}^{n-1}$, une façon intuitive de linéariser cet ensemble est de l'approximer par l'hyperplan qui lui est tangent en x . Cet hyperplan est constitué de l'ensemble des directions orthogonales à x , et on note

$$T_x \mathbb{S}^{n-1} \triangleq \{v \in \mathbb{R}^n : x^\top v = 0\}. \quad (3.1)$$

Pour tout $x \in \mathbb{S}^{n-1}$, cet ensemble est un hyperplan de $\mathbb{R}^n$, donc un espace vectoriel de dimension $n - 1$ (d'où la notation usuelle $\mathbb{S}^{n-1}$, et non $\mathbb{S}^n$). C'est précisément ceci que l'on entend ici par « linéariser *correctement* un sous-ensemble de $\mathbb{R}^n$ » : une approximation locale par un espace euclidien doit être possible en chacun des points de l'ensemble, et la dimension de cet espace ne doit pas varier selon le point choisi. Puisque cette propriété est vérifiée ici, $\mathbb{S}^{n-1}$ sera appelée une *sous-variété lisse plongée dans $\mathbb{R}^n$* . $T_x \mathbb{S}^{n-1}$ est appelé l'*espace tangent à la variété $\mathbb{S}^{n-1}$ en x* . Ceci permet de généraliser la notion de dimension à un ensemble comme $\mathbb{S}^{n-1}$, bien qu'il ne possède pas de structure d'espace vectoriel. La dimension de $\mathbb{S}^{n-1}$ est la dimension commune de ses espaces tangents, c'est-à-dire $n - 1$. Une visualisation de $\mathbb{S}^1$ et de trois de ses espaces tangents est proposée à la figure 3.1(a). Il est à noter que l'équation (3.1) définit bien un espace contenant l'origine de $\mathbb{R}^n$. La visualisation de $T_x \mathbb{S}^{n-1}$ comme tangent à la sphère en x nécessite donc de translater l'espace tangent, ce que nous notons $\{x\} + T_x \Omega_-$. D'autres illustrations présentées dans ce mémoire omettent cette précision.

Mais alors, quels sont les ensembles qui ne peuvent pas être linéarisés de cette façon ? Pour

2. On retrouve dans le choix de ces termes les préoccupations cartographiques aux origines du domaine.

le savoir, il faut d'abord comprendre ce qui a dicté l'intuition derrière l'équation (3.1). Par *fonction lisse*, dans tout ce mémoire, on entend une fonction $\mathcal{C}^\infty$. Supposons que $\mathcal{M}$ est un ensemble non vide pour lequel il existe une fonction lisse $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ ($p \in \mathbb{N}^*$) telle que $\mathcal{M} = \{x \in \mathbb{R}^n : h(x) = 0\}$. Un développement de Taylor de premier ordre autour de $x \in \mathcal{M}$ donne pour $v \in \mathbb{R}^n$:

$$h(x+v) = \underbrace{h(x)}_{=0} + J_h(x)v + O(\|v\|^2)$$

où

$$J_h(x) = \begin{bmatrix} \nabla h_1(x)^\top \\ \nabla h_2(x)^\top \\ \vdots \\ \nabla h_p(x)^\top \end{bmatrix} \in \mathbb{R}^{p \times n}$$

est la matrice jacobienne de h évaluée en x .

On peut ainsi estimer que $x+v$ satisfait presque l'équation définissant $\mathcal{M}$ dès lors que $v \in \text{Ker } J_h(x)$. Il en résulte que l'ensemble des directions linéarisant $\mathcal{M}$ autour de x est donné par $\text{Ker } J_h(x)$. La dimension de cet espace doit être constante pour tout $x \in \mathcal{M}$. Un exemple tiré de [22] permet rapidement de comprendre l'importance de ce critère : étudions l'ensemble $M \subset \mathbb{R}^2$ défini par $h(x) = x_1^2 - x_2^3$. Puisque $J_h(x) = (2x_1, -3x_2^2)$, il apparaît immédiatement que

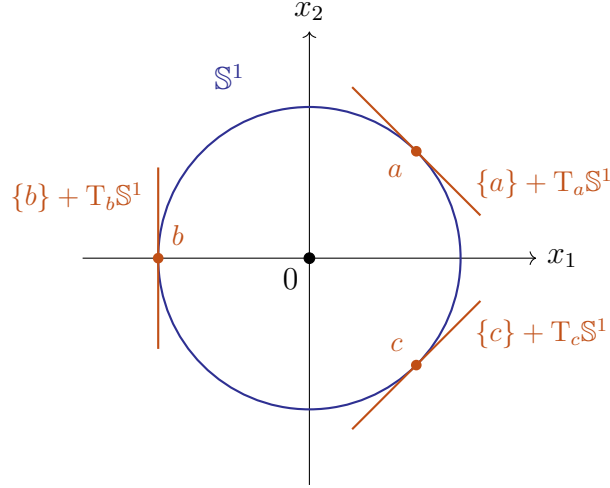
$$\dim(\text{Ker } J_h(x)) = \begin{cases} 1 & \text{si } x \neq 0, \\ 2 & \text{si } x = 0. \end{cases}$$

Autrement dit, M peut être linéarisé par une droite près de chacun de ses points, excepté en l'origine. Puisque $\text{Ker } J_h(0) = \mathbb{R}^2$, la linéarisation de M en l'origine que fournit cette approche est inutile : il s'agit du plan tout entier. Ce constat algébrique traduit la présence d'un point de rebroussement en l'origine, que l'on peut observer à la figure 3.1(b). Puisque M peut être linéarisé par des espaces de dimensions différentes selon le point duquel on part, ce n'est pas une sous-variété lisse de $\mathbb{R}^n$.

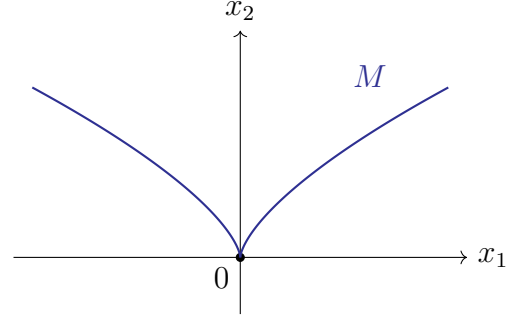
Une fois les bases intuitives posées, vient le temps de définir le type de variétés central dans ce mémoire.

Définition 3.1 (Sous-variété lisse plongée dans $\mathbb{R}^n$ [22, Définition 3.10]). *Soit $\mathcal{M} \subseteq \mathbb{R}^n$ un ensemble non vide. $\mathcal{M}$ est appelé sous-variété lisse de dimension q plongée dans $\mathbb{R}^n$ si l'une des conditions suivantes est vérifiée :*

1. $\mathcal{M}$ est un ouvert de $\mathbb{R}^n$. Dans ce cas, $q = n$.



(a) $\mathbb{S}^1$ et quelques uns de ses espaces tangents.



(b) M n'est pas une sous-variété de $\mathbb{R}^n$.

FIGURE 3.1 Variété, ou non ?

2. Il existe $p \in \mathbb{N}^*$ tel que pour tout $x \in \mathcal{M}$, on puisse trouver un voisinage U de x dans $\mathbb{R}^n$ et une fonction lisse $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ vérifiant :

- (a) pour tout $y \in U$, $y \in \mathcal{M}$ si et seulement si $h(y) = 0$;
- (b) $\text{rang}(J_h(x)) = p$.

Une telle fonction h est appelée fonction localement définissante pour $\mathcal{M}$ en x .

Dans ce cas, $q = n - p$.

On notera $\dim \mathcal{M} \triangleq q$.

Afin de s'épargner cette terminologie précise mais laborieuse, le terme de « sous-variété de $\mathbb{R}^n$ de dimension q » est employé à travers ce mémoire pour désigner de tels ensembles.

La définition plus générale d'une variété, que l'on considère habituellement en géométrie différentielle, nécessite l'introduction d'outils plus élaborés. Les sous-variétés de $\mathbb{R}^n$ en sont des cas particuliers [22, Proposition 8.32].

Bien entendu, la notation h pour désigner une fonction localement définissante pour $\mathcal{M}$ n'est pas choisie au hasard : dans ce mémoire, l'usage fait des sous-variétés suppose que h est un ensemble de contraintes d'égalité. $\mathcal{M}$ sera ainsi l'ensemble réalisable pour ces contraintes.

Les ensembles d'intérêt étant définis, il s'agit maintenant de s'intéresser aux approximations que l'on peut en construire.

3.2 Espaces tangents

La section précédente a proposé une intuition de ce qu'est un espace tangent à une sous-variété $\mathcal{M}$ de $\mathbb{R}^n$ en $x \in \mathcal{M}$, mais il est nécessaire d'en donner une définition plus rigoureuse. Cette définition utilise des courbes paramétrées parcourant $\mathcal{M}$ et passant par x . Plus particulièrement, elle s'intéresse à l'idée de *vitesse* avec laquelle chaque courbe est « lancée » sur la variété.

Définition 3.2 (Espace tangent à une sous-variété de $\mathbb{R}^n$ [22, Définitions 3.14, 3.16]).
Soient $\mathcal{M}$ une sous-variété lisse de $\mathbb{R}^n$ et $x \in \mathcal{M}$. Un vecteur $v \in \mathbb{R}^n$ est dit tangent à $\mathcal{M}$ en x s'il existe un intervalle ouvert $I \subseteq \mathbb{R}$ contenant 0 et une courbe lisse $c : I \rightarrow \mathcal{M}$ telle que $c(0) = x$ et $c'(0) = v$.

L'ensemble de ces vecteurs est appelé l'espace tangent à $\mathcal{M}$ en x , et il est noté $T_x\mathcal{M}$. De plus, x est appelé le point d'ancrage de $T_x\mathcal{M}$ ^a.

a. Le terme *basis* utilisé à cette fin en anglais aurait pu être traduit par « base », et non « point d'ancrage ». Cependant, des bases de $T_x\mathcal{M}$ au sens de la théorie des espaces vectoriels sont utilisées plus loin. Cette terminologie est donc utilisée pour éviter les confusions.

Le caractère abstrait de cette définition contraste avec ce que nous avons manipulé jusqu'à présent ; il est justifié à la section 3.3. Pour le moment, le théorème suivant nous ramène en terrain connu :

Théorème 3.1 ([22, Théorème 3.15]). *Soient $\mathcal{M}$ une sous-variété lisse de $\mathbb{R}^n$ et $x \in \mathcal{M}$. Si $\mathcal{M}$ est un ouvert, alors $T_x\mathcal{M} = \mathbb{R}^n$. Sinon, pour toute fonction localement définissante h pour $\mathcal{M}$ en x (voir définition 3.1), $T_x\mathcal{M} = \text{Ker } J_h(x)$.*

Ceci permet d'exposer le résultat suivant. Bien que simple, il est central dans la méthode développée au chapitre 4.

Proposition 3.2. *Soit $\mathcal{M}$ une sous-variété lisse de $\mathbb{R}^n$ de dimension q . En tout $x \in \mathcal{M}$, $T_x\mathcal{M}$ est isomorphe à $\mathbb{R}^q$ (ce que l'on note $T_x\mathcal{M} \cong \mathbb{R}^q$).*

Démonstration. Ceci découle du point 2b de la définition 3.1, du théorème 3.1 et du fait que tout $\mathbb{R}$ -espace vectoriel de dimension q est isomorphe à $\mathbb{R}^q$. □

L'ensemble des couples constitués d'un point d'ancrage et d'un vecteur tangent associé est regroupé en une dernière définition.

Définition 3.3 (Fibré tangent [22, Définition 3.42]). Soit $\mathcal{M}$ une sous-variété lisse de $\mathbb{R}^n$. On appelle fibré tangent de $\mathcal{M}$ l'ensemble

$$T\mathcal{M} \triangleq \{(x, v) : x \in \mathcal{M}, v \in T_x\mathcal{M}\}.$$

Remarque 3.1. La notation utilisée pour désigner les espaces tangents peut causer des confusions avec certains concepts standards de l'optimisation. Il est donc important de distinguer :

- ▷ $T_X^H(x)$ le cône hypertangent à l'espace $X \subseteq \mathbb{R}^n$ en $x \in X$;
- ▷ $T_x\mathcal{M}$, l'espace tangent à la sous-variété $\mathcal{M}$ de $\mathbb{R}^n$ en $x \in \mathcal{M}$.

Au sens de [56, Définition 12.3], $T_x\mathcal{M}$ est l'ensemble des directions réalisables linéarisées pour les égalités $h(x) = 0$ en $x \in \mathcal{M}$.

Tout espace tangent $T_x\mathcal{M}$ peut être muni du produit scalaire hérité de $\mathbb{R}^n$. Pour clarifier certaines notations, nous définissons

$$\forall v, w \in T_x\mathcal{M}, \quad \langle v, w \rangle_x \triangleq v^\top w,$$

et la norme associée $\|v\|_x = \sqrt{\langle v, v \rangle_x}$ pour tout $v \in T_x\mathcal{M}$. Cette norme induit une métrique sur $T_x\mathcal{M}$ appelée *métrique riemannienne*. Puisque cette construction est possible pour tout $x \in \mathcal{M}$, $\mathcal{M}$ est appelée une sous-variété *riemannienne* de $\mathbb{R}^n$ ³. La topologie induite par cette métrique est équivalente à la topologie induite par $\mathbb{R}^q$ sur $T_x\mathcal{M}$, puisque tous les ensembles étudiés sont de dimensions finies. Afin de clarifier certains énoncés, on définit une notation pour les boules de la topologie induite sur $T_x\mathcal{M}$:

$$\forall v \in T_x\mathcal{M}, \forall \delta > 0, \quad B_x(v, \delta) \triangleq \{w \in T_x\mathcal{M} : \|v - w\|_x < \delta\}. \quad (3.2)$$

3.3 Revenir sur Terre : rétractions

La définition 3.2 indique que tout vecteur tangent à $\mathcal{M}$ en x doit être vu comme la vitesse initiale d'une courbe paramétrée parcourant $\mathcal{M}$ à partir de x . Cette approche est fondamentale : elle incite à voir les directions de $T_x\mathcal{M}$ comme des directions de déplacement sur $\mathcal{M}$ (*via* une courbe) à partir de x . En optimisation riemannienne typiquement, il est désirable que ces directions soient des directions de descente pour un certain objectif $f : \mathcal{M} \rightarrow \mathbb{R}$. Pour autant, si $v \in T_x\mathcal{M}$, il y a fort à parier que $x + v \notin \mathcal{M}$: le dernier outil manquant

3. La définition d'une métrique riemannienne pour $\mathcal{M}$ nécessite en plus que l'application $x \rightarrow \langle \cdot, \cdot \rangle_x$ soit lisse, dans un sens qui est hors du cadre de ce mémoire. Les sous-variétés de $\mathbb{R}^n$ introduites à la définition 3.1 satisfont cette exigence [22, Prop. 3.54].

pour construire un réel moyen de déplacement sur $\mathcal{M}$ est une application $R_x : T_x\mathcal{M} \rightarrow \mathcal{M}$ qui ramène sur $\mathcal{M}$ les déplacements tangents en x . Ces applications doivent faire usage des courbes évoquées ci-dessus.

3.3.1 Pourquoi projeter ne suffit pas

Il semble que le moyen le plus direct de renvoyer le déplacement $v \in T_x\mathcal{M}$ sur $\mathcal{M}$ soit de l'y projeter, par exemple en résolvant le problème suivant :

$$\min_{y \in \mathcal{M}} \|y - (x + v)\| \quad \text{i.e.} \quad \min_{y \in \mathbb{R}^n} \|y - (x + v)\| \quad \text{s.c.} \quad h(y) = 0 \quad (3.3)$$

où h est une fonction localement définissante pour $\mathcal{M}$ en x . Dans le cas où ce problème admet une unique solution, on l'appelle *projection métrique* de $x + v$ sur $\mathcal{M}$, que l'on note $\text{proj}_{\mathcal{M}}(x + v)$.

Du point de vue de la géométrie différentielle, la projection métrique n'est pas entièrement satisfaisante : soit $c : [0, 1] \rightarrow \mathcal{M}$ définie par $c(t) \triangleq \text{proj}_{\mathcal{M}}(x + tv)$. Cette courbe décrit un déplacement sur $\mathcal{M}$ reliant x à $\text{proj}_{\mathcal{M}}(x + v)$, et vérifie $c(0) = x$. Il serait donc désirable qu'elle soit l'une des courbes définissant v comme tangent à $\mathcal{M}$ en x au sens de la définition 3.2. Or cette définition exige que c soit lisse. Il suffit que $\mathcal{M}$ ne soit pas connexe (considérer par exemple un hyperboloïde à deux nappes dans $\mathbb{R}^3$) pour pouvoir rendre une telle courbe discontinue (donc non lisse). Cette observation motive la définition d'une famille d'applications de $T_x\mathcal{M}$ vers $\mathcal{M}$ exploitant les courbes qui possèdent les propriétés désirées. Nous montrons ensuite qu'à une restriction près, la projection métrique fait partie de cette famille.

3.3.2 Rétractions, géodésiques et application exponentielle

Les applications capables de renvoyer un déplacement tangent $v \in T_x\mathcal{M}$ sur $\mathcal{M}$ en suivant les courbes de la définition 3.2 sont appelées des *rétractions*.

Définition 3.4 (Rétraction [22, Définition 3.47]). *Soient $\mathcal{M}$ une sous-variété de $\mathbb{R}^n$ et $x \in \mathcal{M}$. L'application $R_x : T_x\mathcal{M} \rightarrow \mathcal{M}$ est une rétraction sur $\mathcal{M}$ en x si R_x est lisse et si pour tout $v \in T_x\mathcal{M}$, la courbe paramétrée $c(t) \triangleq R_x(tv)$ vérifie $c(0) = x$ et $c'(0) = v$.*

L'application

$$\begin{aligned} R : T\mathcal{M} &\rightarrow \mathcal{M} \\ (x, v) &\mapsto R_x(v) \end{aligned}$$

est une rétraction sur $\mathcal{M}$ si pour tout $x \in \mathcal{M}$, R_x est une rétraction sur $\mathcal{M}$ en x .

Cette définition peut se réinterpréter comme suit : si $(x, v) \in T\mathcal{M}$, l'ensemble des courbes sur $\mathcal{M}$ faisant de v un vecteur tangent au sens de la définition 3.2 et au moins définies sur $[0, 1]$ s'écrit

$$\mathcal{C}_{x,v} \triangleq \{c : I \rightarrow \mathcal{M}, [0, 1] \subseteq I \subseteq \mathbb{R}, c(0) = x, c'(0) = v\}.$$

Une rétraction R_x choisit un élément $c \in \mathcal{C}_{x,v}$ et associe à d le point $R_x(v) \triangleq c(1) \in \mathcal{M}$.

Sur la sphère $\mathbb{S}^{n-1}$, la projection métrique constitue une rétraction, qui consiste à normer le déplacement tangent. Pour tout $x \in \mathbb{S}^{n-1}$, l'application définie par

$$\forall v \in T_x \mathbb{S}^{n-1}, \quad \text{proj}_{\mathbb{S}^{n-1}}(x + v) = \frac{x + v}{\|x + v\|} \quad (3.4)$$

est une rétraction sur $\mathcal{M}$ en x .

Certaines courbes sur $\mathcal{M}$ ont une propriété remarquable : leurs paramétrisations leur confèrent une accélération nulle. Ces courbes sont d'une grande importance : elles généralisent à $\mathcal{M}$ la notion de ligne droite définie dans un espace euclidien. On les appelle les *géodésiques* de $\mathcal{M}$. Il est à noter que la notion d'accélération d'une courbe sur $\mathcal{M}$ n'est pas correctement définie ici, elle nécessiterait des concepts en dehors du cadre de ce mémoire. Nous référons le lecteur intéressé à [22, Théorème 5.29, Définition 5.36] et conservons une approche intuitive de ce concept.

Toute sous-variété de $\mathbb{R}^n$ peut être munie d'une rétraction spéciale qui, parmi les courbes de $\mathcal{C}_{x,v}$, choisit des géodésiques. Cette rétraction est appelée *application exponentielle*.

Définition 3.5 (Application exponentielle). *Soient $x \in \mathcal{M}$ et $v \in T_x \mathcal{M}$. Il existe une unique géodésique $\gamma_v \in \mathcal{C}_{x,v}$ dont le domaine de définition I est maximal (au sens de l'inclusion). Si $[0, 1] \subseteq I$, l'application exponentielle est définie en (x, v) par*

$$\text{Exp}_x(v) \triangleq \gamma_v(1).$$

On note Exp l'application définie sur $T\mathcal{M}$ par $\text{Exp}(x, v) = \text{Exp}_x(v)$.

Il existe des variétés pour lesquelles une expression analytique de Exp est connue. C'est notamment le cas de $\mathbb{S}^{n-1}$, pour laquelle on a :

$$\forall (x, v) \in T\mathbb{S}^{n-1}, \quad \text{Exp}_x(v) = \cos(\|v\|_x) x + \sin(\|v\|_x) \frac{v}{\|v\|_x}. \quad (3.5)$$

Cette application « enroule » l'hyperplan $T_x \mathcal{M}$ autour de la sphère.

La figure 3.2 propose d'illustrer les différences existant entre les rétractions (3.4) et (3.5) sur $\mathbb{S}^2$. Cette figure met en avant les courbes $t \mapsto \text{proj}_{\mathbb{S}^2}(x + tv)$ (en trait plein vert) et $t \mapsto \text{Exp}_x(tv)$ (en trait discontinu rouge) pour $t \in [0, 1]$. Ces courbes se superposent en partie, mais associent chacune à v des points distincts de $\mathcal{M}$. La figure révèle la chose suivante : si ces courbes décrivent le même chemin sur la sphère, elles ne le parcourent pas à la même vitesse. La seconde est une géodésique, ce qui n'est pas le cas de la première. Voici précisément pourquoi la géométrie différentielle ne se contente pas des projections : les courbes qu'elles produisent ne sont pas des géodésiques.

Historiquement, la notion d'application exponentielle précède celle de rétraction. Une définition équivalente à la définition 3.4 est la suivante : une rétraction en x est une approximation de premier ordre de Exp_x [8].

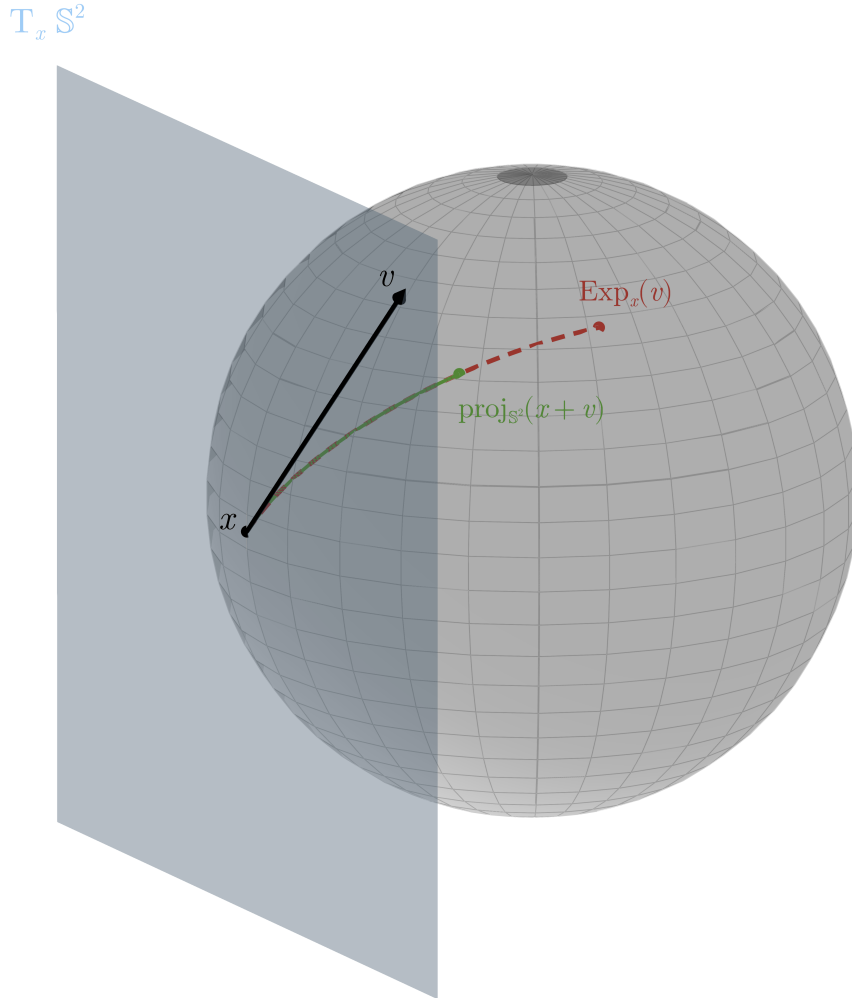


FIGURE 3.2 Deux rétractions distinctes sur $\mathbb{S}^2$

3.3.3 Une projection peut être une rétraction

Bien que l'utilisation de l'application exponentielle soit préférable à celle d'une projection (du moins théoriquement), sa forme analytique est connue pour peu de sous-variétés de $\mathbb{R}^n$. Son approximation numérique est possible, mais requiert la résolution d'un système d'équations différentielles. Recourir à des projections est donc parfois plus intéressant en pratique. Pour l'analyse théorique, il reste cependant essentiel de connaître les conditions sous lesquelles une projection constitue une rétraction, comme c'est le cas pour $\mathbb{S}^{n-1}$.

La proposition suivante établit le fait que proche de l'origine d'un espace tangent, la projection métrique définit une rétraction. Cette proximité de l'origine se caractérise par un voisinage de $0 \in T_x\mathcal{M}$ au sens de la topologie induite sur $T_x\mathcal{M}$.

Proposition 3.3. *Soit $\mathcal{M}$ une sous-variété de $\mathbb{R}^n$. Pour tout $x \in \mathcal{M}$, il existe un voisinage ouvert $\mathcal{V}$ de 0 dans la topologie induite sur $T_x\mathcal{M}$ tel que la restriction*

$$\begin{aligned} \mathcal{V} &\rightarrow \mathcal{M} \\ v &\mapsto \text{proj}_{\mathcal{M}}(x + v) \end{aligned}$$

vérifie les propriétés d'une rétraction.

Démonstration. Ce résultat est en partie démontré dans [22, Prop. 5.54], nous en reprendrons les notations. Soit A l'ensemble des points de $\mathbb{R}^n$ dont la projection métrique sur $\mathcal{M}$ est bien définie, i.e.,

$$a \in A \iff \min_{y \in \mathcal{M}} \|y - a\| \text{ admet une solution unique.}$$

Pour $x \in \mathcal{M}$, on définit

$$\mathcal{O}_x \triangleq \{v \in T_x\mathcal{M} : x + v \in \text{int}(A)\}.$$

Le but est de montrer que $\mathcal{O}_x$ est un voisinage de 0 dans la topologie induite sur $T_x\mathcal{M}$.

$\mathcal{M}$ étant une sous-variété de $\mathbb{R}^n$, un résultat de Federer [39] affirme qu'il existe un voisinage de x dans $\mathbb{R}^n$ inclus dans A . Soit donc $\varepsilon > 0$ tel que $B(x, \varepsilon) \subset A$. On a directement $B(x, \varepsilon) \in \text{int}(A)$.

Pour terminer, il faut exprimer ceci dans la topologie induite sur $T_x\mathcal{M}$. En translatant la boule $B(x, \varepsilon)$ en l'origine de $\mathbb{R}^n$, on obtient :

$$v \in B(0, \varepsilon) \implies x + v \in \text{int}(A).$$

Or l'intersection $B(0, \varepsilon) \cap T_x\mathcal{M}$ est un voisinage ouvert de 0 dans la topologie induite sur

$T_x\mathcal{M}$. Il existe donc $\delta > 0$ tel que $B_x(0, \delta) \subseteq B(0, \varepsilon) \cap T_x\mathcal{M}$. Ainsi,

$$v \in B_x(0, \delta) \implies x + v \in \text{int}(A),$$

i.e., $B_x(0, \delta) \subseteq \mathcal{O}_x$. $B_x(0, \delta)$ étant ouvert dans la topologie induite sur $T_x\mathcal{M}$, elle constitue un voisinage ouvert de l'origine de $T_x\mathcal{M}$ dans lequel la projection métrique vérifie les propriétés d'une rétraction. $\square$

3.4 Algorithmes d'optimisation sur variétés

L'optimisation riemannienne ne s'intéresse pas qu'aux sous-variétés de $\mathbb{R}^n$ introduites à la définition 3.1, mais plus généralement aux sous-variétés d'espaces euclidiens quelconques. Un cas souvent étudié est celui des sous-variétés de $\mathbb{R}^{m \times n}$ ou $\mathbb{C}^{m \times n}$, i.e. des variétés matricielles [7]. Si $\mathcal{M}$ est une sous-variété d'un espace euclidien et $f : \mathcal{M} \rightarrow \mathbb{R}$, un algorithme d'optimisation sur variétés cherche à résoudre le problème

$$\min_{x \in \mathcal{M}} f(x).$$

Grâce à la généralisation des notions de gradient et de hessienne à celles de *gradient riemannien* et de *hessienne riemannienne* [22, Définitions 3.58 et 5.14], des algorithmes standard de l'optimisation avec dérivées (descente de gradient, méthode de Newton, méthodes de région de confiance...) sont généralisés au cas riemannien.

La littérature propose également des algorithmes d'optimisation sans dérivées dans le cas riemannien. Puisque la notion de gradient riemannien généralise celle de gradient euclidien, certaines approches cherchent à reconstituer les gradients riemanniens de f de manière semblable au cas euclidien : par différences finies [65]. Des méthodes de recherche directe ont également été généralisées au cas riemannien. La référence [45] est particulièrement intéressante du point de vue de ce mémoire : à partir d'un itéré $x^k \in \mathcal{M}$, elle considère une rétraction $R_{x^k} : T_{x^k}\mathcal{M} \rightarrow \mathcal{M}$ et recherche dans $T_{x^k}\mathcal{M}$ une direction de descente d pour la composition $f \circ R_{x^k}$. Dès qu'une telle direction est identifiée, l'itération k se termine et l'itéré suivant est calculé par $x^{k+1} \triangleq R_{x^k}(d)$. Cette idée est très proche de ce qui est proposé dans le chapitre 4. Les auteurs généralisent la notion de point raffiné et montrent que de tels points satisfont des conditions nécessaires d'optimalité au sens des dérivées généralisées de Clarke (voir la définition 2.6). Il est également précisé que cette méthode peut être vue comme une généralisation de la méthode des convertisseurs linéaires évoquées à la section 2.3.2. En effet, les sous-espaces affines de $\mathbb{R}^n$ en sont bien des sous-variétés. Une autre référence [26] s'intéresse à des façons efficaces de générer des directions engendrant positivement un espace tangent

$T_x\mathcal{M}$ dans une telle méthode. Elle s'intéresse également à la complexité de ces méthodes.

Certaines références étudient précisément l'adaptation de MADS au cas riemannien. Le cas des sphères est abordé dans [64]. Les références [35, 36, 37, 34] constituent une inspiration majeure pour le travail réalisé ici : elles sont les premières à suggérer de résoudre avec une méthode de recherche directe comme MADS, des sous-problèmes successifs sur les espaces tangents. L'auteur a recours à l'application exponentielle sur $\mathcal{M}$ afin de résoudre, pour $x^k \in \mathcal{M}$, le sous-problème

$$\begin{aligned} \min_{v \in T_{x^k}\mathcal{M}} \quad & f \circ \text{Exp}_{x^k}(v) \\ \text{s.c.} \quad & g \circ \text{Exp}_{x^k}(v) \leq 0. \end{aligned} \tag{3.6}$$

Les pas autorisés pour la recherche directe sont limités à la zone dans laquelle Exp_{x^k} est inversible. Cette idée est reprise dans la section 4.1.2. Sous cette hypothèse, l'auteur affirme que les résultats de convergence dont est munie la méthode de recherche directe sont préservés. Ces travaux offrent de nombreux développements théoriques sur lesquels la démarche du chapitre 4 est appuyée. En particulier, l'auteur affirme dans [36] que cette approche est pertinente pour le traitement des problèmes contraints par égalités. La contribution de ce mémoire peut être comprise comme une généralisation et une mise en pratique de cette méthode. Ces travaux ne proposent en revanche aucun test numérique.

Une implémentation de la méthode décrite dans [35] sans contraintes d'inégalité est effectuée dans le cadre du travail [57]. Une instance de *Lower-Triangular Mesh Adaptive Direct Search* (LTMADS) est implémentée pour résoudre le problème (3.6) sur des variétés pour lesquelles l'application exponentielle est connue. Si la phase de recherche identifie un point $\hat{v} \in T_{x^k}\mathcal{M}$ améliorant la valeur de $f \circ \text{Exp}_{x^k}$, l'algorithme se déplace en $x^{k+1} \triangleq \text{Exp}_{x^k}(\hat{v})$. Le treillis utilisé pour résoudre le problème dans $T_{x^k}\mathcal{M}$ est alors transformé par une opération appelée *transport vectoriel*, et forme le treillis de départ pour le nouveau sous-problème à résoudre dans $T_{x^{k+1}}\mathcal{M}$. Cet algorithme est utilisé pour résoudre le problème dit *de Procrustes* : pour deux matrices $A, B \in \mathbb{R}^{n \times m}$, il s'agit d'identifier une matrice orthogonale $X \in \mathbb{R}^{n \times n}$ minimisant la quantité $\|XA - B\|$ (le choix de la norme matricielle définit une variante du problème). L'ensemble des matrices $n \times n$ orthogonales est une sous-variété riemannienne de $\mathbb{R}^{n \times n}$, appelée *groupe de rotation d'ordre n* . Cette méthode est implémentée dans la librairie  **Manopt.jl** du langage Julia [4], nous l'utiliserons au chapitre 5 pour comparer ses performances à celles de notre méthode.

Tous les outils nécessaires étant à notre disposition, il est temps de présenter la contribution principale de ce mémoire : la méthode de *Riemannian Derivative-Free Optimization*.

CHAPITRE 4 CONCEPTION D'UNE MÉTHODE Ω_- -RÉALISABLE SUR VARIÉTÉS RIEMANNIENNES

L'exemple de la section 2.1.4 montre qu'en présence d'égalités, la théorie de convergence d'un algorithme utilisant la notion de cône hypertangent est mise à mal par la géométrie du domaine Ω_- . Dans le cas où cette géométrie est connue, il est possible d'imaginer une nouvelle approche visant à « guider » un algorithme d'optimisation sans dérivées dans ce domaine, afin de produire une suite d'itérés Ω_- -réalisables. Ceci présuppose de sortir quelque peu du cadre sans dérivées – du moins en ce qui concerne les contraintes d'égalité – à l'instar des méthodes évoquées dans les sections 2.3.2 et 3.4.

Un premier obstacle survient lorsque l'on considère des contraintes d'égalité non linéaires : dans la plupart des cas, l'ensemble Ω_- ne possède plus de structure euclidienne. Notamment, il n'est plus stable ni par addition ni par multiplication scalaire, deux opérations fondamentales pour la plupart des algorithmes d'optimisation, avec ou sans dérivées.

Pour adapter MADS à un tel contexte, il serait envisageable de construire l'équivalent d'un treillis courbé sur un espace non euclidien. Nous y préférons une autre approche similaire aux travaux évoqués dans la section 3.4 : en supposant que Ω_- constitue une sous-variété riemannienne de $\mathbb{R}^n$ de dimension q , ses espaces tangents permettent de formuler des sous-problèmes de dimension q et ainsi, agir sur un élément crucial pour les méthodes de recherche directionnelle : la taille des problèmes à résoudre.

Ce chapitre introduit une méthode qui exploite cette démarche à l'aide d'un solveur sans dérivées quelconque. Cette méthode est nommée RDFO pour *Riemannian Derivative-Free Optimization*. Nous en développons le principe et l'analyse de convergence.

Dans tout ce chapitre, il est fait usage des concepts et des notations introduits au chapitre 3.

4.1 Principes et outils de RDFO

L'algorithme RDFO traite des instances de (P) satisfaisant l'hypothèse suivante.

Hypothèse 4.1. *L'ensemble $\{x \in \mathbb{R}^n : h(x) = 0\}$ est une sous-variété riemannienne de $\mathbb{R}^n$ au sens de la définition 3.1. On note $\mathcal{M} \triangleq \{x \in \mathbb{R}^n : h(x) = 0\}$ et $q \triangleq \dim \mathcal{M}$.*

Cette section expose progressivement les idées, outils et notations nécessaires à la présentation de RDFO. Le symbole $\mathcal{M}$ est systématiquement utilisé pour désigner une sous-variété

riemannienne de $\mathbb{R}^n$ de dimension $q \in \mathbb{N}^*$.

4.1.1 Sous-problèmes dans $T\mathcal{M}$

L'idée centrale de RDFO étend celle proposée dans [35] à l'utilisation de rétractions différentes de l'application exponentielle. Il s'agit donc toujours de formuler des sous-problèmes successifs dans des espaces tangents à $\mathcal{M}$, et de rétracter les solutions obtenues afin de générer une suite d'itérés Ω_- -réalisables.

Si $x \in \mathcal{M}$ et R_x est une rétraction sur $\mathcal{M}$ en x , notons $f_x \triangleq f \circ R_x$ et $g_x \triangleq g \circ R_x$. Le sous-problème sur $T_x\mathcal{M}$ généralisant (3.6) est :

$$\begin{aligned} \min_{v \in T_x\mathcal{M}} \quad & f_x(v) \\ \text{s.c.} \quad & g_x(v) \leq 0. \end{aligned} \tag{4.1}$$

L'avantage de cette approche réside dans l'utilisation de la proposition 3.2, qui révèle que ce sous-problème est en réalité de dimension q . Sa reformulation explicite sera détaillée au chapitre 5.

Le lien entre une solution du sous-problème (4.1) et une solution de (P) dépend des propriétés de R_x .

4.1.2 Rétractions et rayon d'inversibilité

Le choix d'une rétraction est un élément déterminant dans l'utilisation d'un algorithme d'optimisation riemannienne. Rappelons ce qui a été énoncé aux sections 3.3.2 et 3.3.3 : au sens de la géométrie différentielle, certaines rétractions ont des propriétés plus désirables que d'autres ; c'est notamment le cas de l'application exponentielle. Ces propriétés viennent avec une contrepartie, à savoir la difficulté, pour la plupart des variétés, à déterminer Exp_x sous une forme analytique ou numérique. La proposition 3.3 permet de se reposer sur la projection métrique, qui satisfait les propriétés d'une rétraction au moins localement autour de l'origine d'un espace tangent.

Dans la formulation du sous-problème (4.1), l'espoir est que la rétraction utilisée décrive au mieux la correspondance qui existe entre $T_x\mathcal{M}$ et $\mathcal{M}$ au voisinage du point x , i.e., qu'elle soit *invertible*. La proposition suivante affirme que tel est le cas localement, dans un voisinage de l'origine de $T_x\mathcal{M}$ au sens de la topologie induite (dans le même esprit que la proposition 3.3). Rappelons que l'on entend par *difféomorphisme* une application inversible de classe $\mathcal{C}^1$, dont la réciproque est également de classe $\mathcal{C}^1$.

Proposition 4.1. *Soient $\mathcal{M}$ une sous-variété de $\mathbb{R}^n$ et $x \in \mathcal{M}$. Soit $R_x : T_x \mathcal{M} \rightarrow \mathcal{M}$ une rétraction sur $\mathcal{M}$ en x . Il existe un voisinage ouvert $\mathcal{V}$ de $0 \in T_x \mathcal{M}$ tel que la restriction*

$$\begin{aligned} \mathcal{V} &\rightarrow \mathcal{M} \\ v &\mapsto R_x(v) \end{aligned}$$

soit un difféomorphisme sur son image.

Démonstration. Ce résultat découle d’une définition alternative d’une rétraction et du théorème de la fonction inverse sur variétés [22, Th. 4.16]. $\square$

La taille de ce voisinage dépend bien sûr de la rétraction étudiée. Dans le cas de l’application exponentielle, elle est connue sous le nom de *rayon d’injectivité* de $\mathcal{M}$ en x . Avec la notation définie en (3.2), le rayon d’injectivité de $\mathcal{M}$ en x s’écrit

$$\text{inj}(x) \triangleq \sup\{\delta > 0 \mid \text{Exp}_x : B(0, \delta) \rightarrow \mathcal{M} \text{ est un difféomorphisme sur son image}\}.$$

Il est par exemple connu qu’en tout point $x \in \mathbb{S}^{n-1}$, on a $\text{inj}(x) = \pi$, et ce quel que soit $n \in \mathbb{N}^*$. Si $d \in T_x \mathbb{S}^{n-1}$ est tel que $\|d\|_x = 1$, Exp_x crée donc une bijection entre le segment $[0, \pi d] \subset T_x \mathbb{S}^{n-1}$ et un demi grand cercle reliant x à son antipode sur la sphère (de même avec le segment $[-\pi d, 0]$).

Puisque la proposition précédente affirme que toute rétraction est localement inversible, elle motive la généralisation du concept de rayon d’injectivité à une rétraction quelconque. Nous proposons de nommer cette généralisation le *rayon d’inversibilité*.

Définition 4.1 (Rayon d’inversibilité). *Soient $\mathcal{M}$ une sous-variété de $\mathbb{R}^n$, $x \in \mathcal{M}$ et R_x une rétraction sur $\mathcal{M}$ en x . On appelle rayon d’inversibilité de R_x la quantité*

$$\text{inv}(R_x) \triangleq \sup\{\delta > 0 \mid R_x : B_x(0, \delta) \rightarrow \mathcal{M} \text{ est un difféomorphisme sur son image}\}.$$

En vertu de la proposition 4.1, $\text{inv}(R_x) > 0$, pour tout $x \in \mathcal{M}$.

Malheureusement, déterminer le rayon d’inversibilité d’une rétraction arbitraire est une tâche difficile. Dans le cadre de l’algorithme RDFO, connaître une borne inférieure sur ce rayon suffit à conserver un contrôle sur la résolution des sous-problèmes. Nous introduisons une notation dédiée : dans le contexte où R_x est une rétraction sur $\mathcal{M}$ en x , on note $\rho(R_x)$ une quantité vérifiant $\rho(R_x) \leq \text{inv}(R_x)$.

4.1.3 Idée et notations de RDFO

L'idée de l'algorithme RDFO est de résoudre successivement des sous-problèmes de la forme de (4.1) pour plusieurs points d'ancrages $x \in \mathcal{M}$ à l'aide d'un solveur sans dérivées. L'algorithme se termine lorsque le solveur réussit à converger dans la région où R_x est inversible. La même utilisation du rayon d'injectivité de $\mathcal{M}$ est faite dans [34].

Les points d'ancrage successifs sont appelés les *itérés externes* de RDFO, indicés par $\ell \in \mathbb{N}$. On note ainsi $x_\ell \in \mathcal{M}$ le $\ell^{\text{ième}}$ itéré externe de RDFO. Cet itéré représente la meilleure solution Ω_- -réalisable obtenue par l'algorithme au bout de ℓ itérations.

Avec cette notation, on propose l'écriture suivante (finale) des sous-problèmes résolus par RDFO : le sous-problème associé à $x_\ell \in \mathcal{M}$ est

$$\begin{aligned} \min_{v \in T_x \mathcal{M}} \quad & f_{x_\ell}(v) \\ \text{s.c.} \quad & g_{x_\ell}(v) \leq 0. \end{aligned} \tag{SP}_\ell$$

L'algorithme présenté à la section suivante introduit la notation v_ℓ^* , pour désigner le meilleur point trouvé par le solveur lors de la résolution de (SP_ℓ) . Cette notation illustre le principe utilisé dans RDFO : en indice, on place toujours la référence aux itérations externes : x_ℓ est un itéré externe (situé sur $\mathcal{M}$), v_ℓ^* est la solution obtenue lors de la résolution du $\ell^{\text{ième}}$ sous-problème. Au même titre, la solution finale de (P) fournie par RDFO est notée $x_\star$, car elle est optimale pour (P). Les exposants sont réservés au statut des éléments vis-à-vis des sous-problèmes : dans v_ℓ^* , le symbole $*$ est porté en exposant, puisque cet élément est le meilleur obtenu dans le cadre du $\ell^{\text{ième}}$ sous-problème.

4.2 L'algorithme

RDFO repose sur l'hypothèse 4.1, ainsi que sur la suivante :

Hypothèse 4.2. *Pour tout $x \in \mathcal{M}$, une rétraction R_x ainsi qu'une borne inférieure sur son rayon d'inversibilité $\rho(R_x) \leq \text{inv}(R_x)$ peuvent être calculées.*

Avec ces hypothèses, la méthode est présentée formellement à l'algorithme 4.1.

Il est à noter que nous utilisons le même vocabulaire que les algorithmes de recherche directe directionnelle. En particulier, à la ligne 6, l'expression « un itéré v tel que $\|v\|_{x_\ell} \geq \rho(R_{x_\ell})$ » désigne un point v retenu par l'algorithme comme plus favorable que la solution courante. Ainsi, la résolution du sous-problème (SP_ℓ) est interrompue dès lors qu'une amélioration est

Algorithme 4.1 : RDFO : Optimisation riemannienne sans dérivées

Données : $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $x_{\text{init}} \in \mathbb{R}^n$, un solveur sans dérivées capable de résoudre (SP_ℓ) et les objets définis dans les hypothèses 4.1 et 4.2.

¹ **0.** Initialisation

² $x_0 \leftarrow \text{proj}_{\mathcal{M}}(x_{\text{init}})$
³ $\ell \leftarrow 0$

⁴ **1.** Résolution du sous-problème

⁵ Appliquer le solveur sans dérivées à (SP_ℓ) à partir de $v^0 = 0$.

⁶ **si** le solveur génère un itéré v tel que $\|v\|_{x_\ell} \geq \rho(\text{R}_{x_\ell})$ **alors**

⁷ Interrompre le solveur.

⁸ $v_\ell^* \leftarrow$ le premier itéré tel que $\|v^*\|_{x_\ell} \geq \rho(\text{R}_{x_\ell})$

⁹ Aller en 2

¹⁰ **sinon**

¹¹ $v_\ell^* \leftarrow$ la solution identifiée par le solveur

¹² $x_\star \leftarrow \text{R}_{x_\ell}(v_\ell^*)$

¹³ Fin.

¹⁴ **2.** Mises à jour

¹⁵ $x_{\ell+1} \leftarrow \text{R}_{x_\ell}(v_\ell^*)$

¹⁶ $\ell \leftarrow \ell + 1$

¹⁷ Aller en 1.

observée en dehors du rayon d'inversibilité de R_x , et non simplement dès que le solveur évalue un point en dehors de ce rayon.

4.3 Analyse de convergence de RDFO avec MADS

Certains solveurs sans dérivées garantissent, dans le cas où ils convergent, que les solutions qu'ils retournent satisfont des conditions nécessaires d'optimalité. C'est notamment le cas de MADS, pour lequel ces conditions ont été énoncées au théorème 2.2. Cette section est dédiée à énoncer et prouver des conditions nécessaires que satisfait une solution produite par une instance de RDFO dans laquelle MADS est utilisé comme solveur sans dérivées. Le cadre de l'analyse de convergence est donc le cas où l'algorithme s'arrête après un nombre fini d'itérations extérieures. Notons que nous ne nous intéressons pas ici aux conditions nécessaires pour que cette convergence ait lieu.

Dans le cas où RDFO s'arrête après un nombre fini d'itérations extérieures, il existe un indice $\ell \in \mathbb{N}$ pour lequel la résolution du sous-problème (SP_ℓ) a été menée jusqu'à la convergence théorique du solveur sans dérivées. L'idée de l'analyse qui suit est d'étudier les conditions d'optimalité satisfaites dans $T_{x_\ell}\mathcal{M}$ par la solution v_ℓ^* de ce sous-problème, et de tenter de les transférer dans l'espace tangent $T_{x_\star}\mathcal{M}$ où $x_\star = R_{x_\ell}(v_\ell^*)$.

Dans toute cette section, on notera donc $\ell \in \mathbb{N}$ l'indice du dernier sous-problème résolu par la convergence du solveur sans dérivées, et $x_\star = R_{x_\ell}(v_\ell^*)$.

4.3.1 Convergence de MADS sur le dernier sous-problème

L'analyse de convergence débute en énonçant le résultat de convergence de MADS dans $T_{x_\ell}\mathcal{M}$.

Lemme 4.2. *Dans le cas où l'algorithme 4.1 est exécuté avec MADS pour solveur et s'arrête après un nombre fini d'itérations extérieures, soient $\ell \in \mathbb{N}$ l'indice de la dernière de ces itérations et v_ℓ^* la solution identifiée de (SP_ℓ) . Il existe un ensemble infini d'indices $K \subseteq \mathbb{N}$ et une sous-suite raffinant $\{v^k\}_{k \in K}$ générée par MADS lors de la résolution de (SP_ℓ) vérifiant*

$$\lim_{k \in K} v^k = v_\ell^*.$$

De plus,

▷ il existe une suite associée $\{d^k\}_{k \in K}$ de directions de sonde telle que

$$d^* \triangleq \lim_{k \in K} \frac{d^k}{\|d^k\|}$$

est une direction raffinante ;

▷ la suite associée $\{\delta^k\}_{k \in K}$ des paramètres de taille de treillis vérifie

$$\lim_{k \in K} \delta^k = 0$$

et

$$\limsup_{k \in K} \frac{f_{x_\ell}(v^k + \delta^k d^k) - f_{x_\ell}(v^k)}{\delta^k} \geq 0.$$

Démonstration. Ce résultat découle du fait que v_ℓ^* est un point raffiné produit par MADS. Les suites $\{d^k\}$ et $\{\delta^k\}$ sont celles impliquées dans la définition 2.5. Cette même définition implique que

$$\forall k \in K, \quad f_{x_\ell}(v^k) \leq f_{x_\ell}(v^k + \delta^k d^k).$$

Puisque $\delta^k > 0$ pour tout $k \in K$, le passage à la limite supérieure est immédiat. $\square$

Une fois ce résultat énoncé, nous nous intéressons aux conséquences que l'on peut lui trouver dans $T_{x_\star} \mathcal{M}$.

4.3.2 Opération de transfert

L'analyse du résultat du lemme 4.2 du point de vue de $T_{x_\star} \mathcal{M}$ nécessite de pouvoir passer de $T_{x_\ell} \mathcal{M}$ à $T_{x_\star} \mathcal{M}$. Nous introduisons pour cela l'opération suivante, dite de *transfert*.

Définition 4.2 (Opération de transfert). *Soient $x_1, x_2 \in \mathcal{M}$ et R_{x_1}, R_{x_2} des rétractions sur $\mathcal{M}$, respectivement en x_1 et en x_2 . Si $v \in T_{x_1} \mathcal{M}$ est tel que $R_{x_1}(v) \in R_{x_2}(B_{x_2}(0, \text{inv}(R_{x_2})))$, alors le point $R_{x_2}^{-1} \circ R_{x_1}(v) \in T_{x_2} \mathcal{M}$ est appelé le transfert de v de $T_{x_1} \mathcal{M}$ vers $T_{x_2} \mathcal{M}$.*

L'opération transfère le point v en le rétractant depuis $T_{x_1} \mathcal{M}$, puis en ramenant le résultat de cette rétraction sur $T_{x_2} \mathcal{M}$ en y appliquant $R_{x_2}^{-1}$. Cette seconde opération est souvent appelée *lifting*. La condition $R_{x_1}(v) \in R_{x_2}(B_{x_2}(0, \text{inv}(R_{x_2})))$ assure simplement que le *lifting* de $R_{x_1}(v)$ sur $T_{x_2} \mathcal{M}$ est bien défini. L'opération entière peut se comprendre comme un passage de $T_{x_1} \mathcal{M}$ à $T_{x_2} \mathcal{M}$ après avoir « rebondi » sur $\mathcal{M}$.

Cette opération ne coïncide ni avec la notion de *transport parallèle*, fondamentale en géométrie

différentielle [22, Définition 10.35], ni avec celle de transport vectoriel [7, Définition 8.1.1]. Elle est également à ne pas confondre avec la notion de *transporteur* [22, Définition 10.61], utilisée dans [57]. L'utilisation de ces outils serait à propos ici, et pourrait fournir de nouveaux résultats. Nous leur préférons cependant des résultats obtenus uniquement à partir des outils utilisés jusqu'ici, à savoir les rétractions.

Le résultat de convergence que nous visons concerne les dérivées généralisées de Clarke de $f_{x_\star}$. Mais en quel point ces dérivées doivent-elles être étudiées ? Puisque MADS a convergé lors de la résolution de (SP_ℓ) , le théorème 2.3 affirme que

$$f_{x_\ell}^\circ(v_\ell^*; d^*) \geq 0,$$

où d^* est une direction raffinant telle que $d^* \in T_{\Omega_\leq}^H(x_\ell)$. Nous pouvons donc nous demander ce qu'il advient du point v_ℓ^* après transfert dans $T_{x_\star}\mathcal{M}$. Par construction :

$$R_{x_\star}^{-1} \circ R_{x_\ell}(v_\ell^*) = R_{x_\star}^{-1}(x_\star) = 0. \quad (4.2)$$

Il est donc naturel que notre analyse de convergence s'intéresse aux dérivées de Clarke de $f_{x_\star}$ au point 0. Si $w \in T_{x_\star}\mathcal{M}$, on a :

$$f_{x_\star}^\circ(0; w) = \limsup_{\substack{y \rightarrow 0 \\ t \searrow 0}} \frac{f_{x_\star}(y + tw) - f_{x_\star}(y)}{t}. \quad (4.3)$$

Calculer une telle dérivée nécessite d'étudier des suites de points de $T_{x_\star}\mathcal{M}$ convergeant vers son origine. L'opération de transfert permet d'établir, au voisinage de $0 \in T_{x_\star}\mathcal{M}$, une correspondance exacte entre ces suites et celles qui convergent vers v_ℓ^* dans $T_{x_\ell}\mathcal{M}$.

Lemme 4.3. *Soient $x_1 \in \mathcal{M}$, $v \in T_{x_1}\mathcal{M}$ tel que $\|v\|_{x_1} < \text{inv}(R_{x_1})$ et $x_2 \triangleq R_{x_1}(v) \in \mathcal{M}$. Soit $(y^i)_{i \in \mathbb{N}}$ une suite d'éléments de $T_{x_2}\mathcal{M}$. La suite (y^i) converge vers $0 \in T_{x_2}\mathcal{M}$ si et seulement s'il existe un indice $I \in \mathbb{N}$ tel que :*

$$\forall i \geq I, \exists v^i \in T_{x_1}\mathcal{M} : y^i = R_{x_2}^{-1} \circ R_{x_1}(v^i)$$

et que $\lim_{i \rightarrow \infty} v^i = v$.

Démonstration. Cet énoncé se prouve en identifiant des voisinages ouverts de $v \in T_{x_1}\mathcal{M}$ et de $0 \in T_{x_2}\mathcal{M}$ dans lesquels l'opération de transfert reste un difféomorphisme.

La restriction de R_{x_1} à $B_{x_1}(0, \text{inv}(R_{x_1}))$ sur son image étant un difféomorphisme, cette appli-

cation envoie les ouverts de $B_{x_1}(0, \text{inv}(\mathbf{R}_{x_1}))$ sur des ouverts au sens de la topologie de $\mathcal{M}$. Puisque $x_2 = \mathbf{R}_{x_1}(v)$ et que $\|v\|_{x_1} < \text{inv}(\mathbf{R}_{x_1})$, il existe un voisinage ouvert de x_2 au sens de la topologie de $\mathcal{M}$, dans lequel $\mathbf{R}_{x_1}$ est un difféomorphisme. Soit $\mathcal{U}_1$ ce voisinage.

Le même raisonnement peut être répété avec $0 \in \mathbf{T}_{x_2}\mathcal{M}$ afin d'obtenir un second voisinage ouvert $\mathcal{U}_2$ de x_2 au sens de la topologie de $\mathcal{M}$, dans lequel $\mathbf{R}_{x_2}$ est un difféomorphisme.

Soit alors $\mathcal{U}$ le plus petit des voisinages $\mathcal{U}_1$ et $\mathcal{U}_2$ au sens de l'inclusion. Les ensembles $\mathbf{R}_{x_1}^{-1}(\mathcal{U})$ et $\mathbf{R}_{x_2}^{-1}(\mathcal{U})$ sont des voisinages ouverts de $v \in \mathbf{T}_{x_1}\mathcal{M}$ et de $0 \in \mathbf{T}_{x_2}\mathcal{M}$ dans lesquels $\mathbf{R}_{x_1}$ et $\mathbf{R}_{x_2}$ sont des difféomorphismes, respectivement. Il en résulte que l'opération de transfert $\mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}$ est une composition de difféomorphismes, i.e., un difféomorphisme, de $\mathbf{R}_{x_1}^{-1}(\mathcal{U})$ dans $\mathbf{R}_{x_2}^{-1}(\mathcal{U})$.

De ceci, on déduit :

$\Leftarrow$ Soit $I \in \mathbb{N}$ un indice à partir duquel $i \geq I$ implique qu'il existe $v^i \in \mathbf{T}_{x_1}\mathcal{M}$ tel que $y^i = \mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}(v^i)$. Si $\lim_{i \rightarrow \infty} v^i = v$ et puisque $\mathbf{R}_{x_1}^{-1}(\mathcal{U})$ est un voisinage ouvert de v dans $\mathbf{T}_{x_1}\mathcal{M}$ défini précédemment, il existe $\tilde{I} \in \mathbb{N}$ tel que si $i \geq \tilde{I}$, alors $v^i \in \mathbf{R}_{x_1}^{-1}(\mathcal{U})$. Soit $\hat{I} \triangleq \max\{I, \tilde{I}\}$. Pour tout $i \geq \hat{I}$, $\mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}$ est un difféomorphisme au voisinage de v^i . Il est donc en particulier continu, ce qui permet d'écrire :

$$\lim_{i \rightarrow \infty} y^i = \lim_{i \rightarrow \infty} \mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}(v^i) = \mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1} \left(\lim_{i \rightarrow \infty} v^i \right) = \mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}(v) = \mathbf{R}_{x_2}^{-1}(x_2) = 0.$$

$\Rightarrow$ Si $\lim_{i \rightarrow \infty} y^i = 0$ et puisque $\mathbf{R}_{x_2}^{-1}(\mathcal{U})$ est un voisinage ouvert de d dans $\mathbf{T}_{x_2}\mathcal{M}$ défini précédemment, il existe $I \in \mathbb{N}$ tel que $i \geq I \implies y^i \in \mathbf{R}_{x_2}^{-1}(\mathcal{U})$. Puisque $\mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}$ est un difféomorphisme sur cet ensemble, il existe $v^i \in \mathbf{R}_{x_1}^{-1}(\mathcal{U}) \subseteq \mathbf{T}_{x_1}\mathcal{M}$ tel que $y^i = \mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1}(v^i)$. Par continuité de ce difféomorphisme (et de son inverse), on obtient :

$$\begin{aligned} \lim_{i \rightarrow \infty} v^i &= \lim_{i \rightarrow \infty} (\mathbf{R}_{x_2}^{-1} \circ \mathbf{R}_{x_1})^{-1}(y^i) \\ &= \lim_{i \rightarrow \infty} \mathbf{R}_{x_1}^{-1} \circ \mathbf{R}_{x_2}(y^i) = \mathbf{R}_{x_1}^{-1} \circ \mathbf{R}_{x_2} \left(\lim_{i \rightarrow \infty} y^i \right) = \mathbf{R}_{x_1}^{-1} \circ \mathbf{R}_{x_2}(0) = \mathbf{R}_{x_1}^{-1}(x_2) = v. \end{aligned}$$

□

Ce lemme établit un lien clé pour notre analyse entre les espaces $\mathbf{T}_{x_\ell}\mathcal{M}$ et $\mathbf{T}_{x_\star}\mathcal{M}$. Il reste à déterminer les directions $w \in \mathbf{T}_{x_\star}\mathcal{M}$ dans lesquelles la dérivée de Clarke (4.3) doit être étudiée.

4.3.3 Résultat de convergence

Afin de choisir une direction dans laquelle étudier la dérivée de Clarke de $f_{x_\star}$, nous étudions les suites définies au lemme 4.2 : soient $\{v^k\}_{k \in K}$ une sous-suite raffinante produite par MADS, d^* une direction raffinante produite par les directions de sonde $\{d^k\}_{k \in K}$ telle que $R_{x_\ell}(d^*) \in T_{\Omega_\leq}^H(x_\ell)$ et $\{\delta^k\}_{k \in K}$ la suite des paramètres de taille de treillis associée. La figure 4.1(a) représente une configuration possible de ces suites dans $T_{x_\ell}\mathcal{M}$, et attire l'attention sur la direction $\delta^k d^k$: cette suite de directions est précisément celle qui rend $f_{x_\ell}^\circ(v^*; d^*)$ positive. Nous étudions donc ce qu'il advient de ces directions après transfert dans $T_{x_\star}\mathcal{M}$, en introduisant la suite $\{w^k\}_{k \in K}$ définie par

$$w^k = \frac{R_{x_\star}^{-1} \circ R_{x_\ell}(v^k + \delta^k d^k) - R_{x_\star}^{-1} \circ R_{x_\ell}(v^k)}{\delta^k}.$$

La figure 4.1(b) illustre le transfert de la suite $\{v^k\}$ dans $T_{x_\star}\mathcal{M}$, et l'image du déplacement par ce transfert, qui correspond à $\delta^k w^k$. Les directions w^k sont les images (mises à l'échelle) des directions de sonde dans $T_{x_\ell}\mathcal{M}$; les directions qui doivent occuper notre analyse de convergence sont donc les directions d'accumulation de $\{w^k\}$.

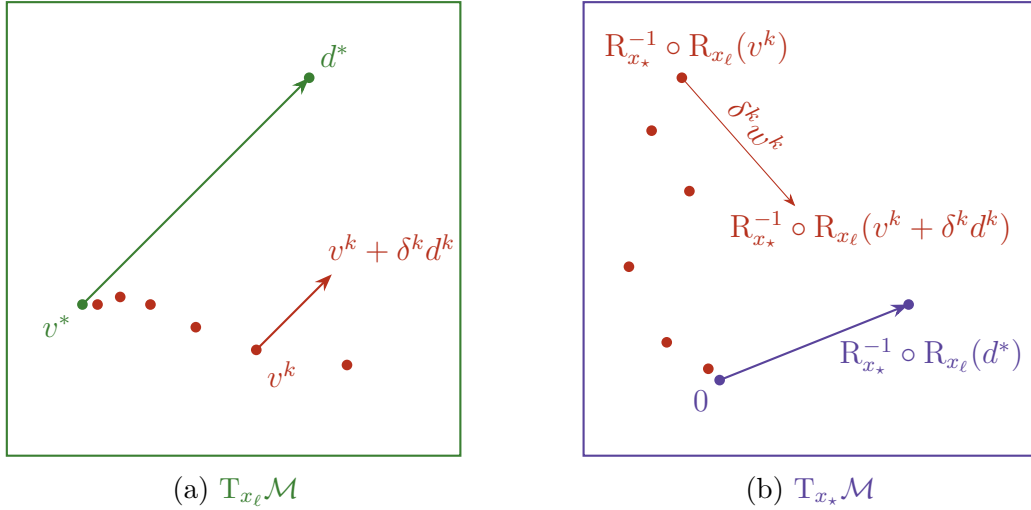


FIGURE 4.1 Illustration des suites impliquées dans la preuve du théorème 4.4.

Théorème 4.4 (Condition nécessaire d'optimalité). *Dans le cas où l'algorithme 4.1 est exécuté avec MADS pour solveur et s'arrête après un nombre fini d'itérations extérieures, soit ℓ l'indice de la dernière de ces itérations. Soient les suites $\{v^k\}$, $\{d^k\}$ et $\{\delta^k\}$ définies*

au lemme 4.2 et $\hat{w}$ un point d'accumulation de la suite $\{w^k\}_{k \in K}$ définie par

$$w^k = \frac{R_{x_\star}^{-1} \circ R_{x_\ell}(v^k + \delta^k d^k) - R_{x_\star}^{-1} \circ R_{x_\ell}(v^k)}{\delta^k}.$$

Supposons que $d^* \triangleq \lim_{k \in K} d^k$ soit tel que $R_{x_\ell}(d^*) \in T_{\Omega_\leq}^H(x_\ell)$. Si f est lipschitzienne au voisinage de $x_\star$ et possède des ensembles de sous-niveau bornés, alors

$$f_{x_\star}^\circ(0; \hat{w}) \geq 0.$$

Démonstration. La dérivée directionnelle de Clarke qui nous occupe est

$$f_{x_\star}^\circ(0; \hat{w}) = \limsup_{\substack{y^j \rightarrow 0 \\ t^j \searrow 0}} \frac{f_{x_\star}(y^j + t^j \hat{w}) - f_{x_\star}(y^j)}{t^j}.$$

Puisque $x_\star = R_{x_\ell}(v_\ell^*)$, le lemme 4.3 permet de réécrire toute suite $y^j \rightarrow 0$ dans $T_{x_\star} \mathcal{M}$ sous la forme $y^j = R_{x_\star}^{-1} \circ R_{x_\ell}(v^j)$ à partir d'un certain indice, avec $v^j \in T_{x_\ell} \mathcal{M}$ et $v^j \rightarrow v_\ell^*$. On obtient ainsi

$$f_{x_\star}^\circ(0; \hat{w}) = \limsup_{\substack{v^j \rightarrow v_\ell^* \\ t^j \searrow 0}} \frac{f_{x_\star}(R_{x_\star}^{-1} \circ R_{x_\ell}(v^j) + t^j \hat{w}) - f_{x_\star}(R_{x_\star}^{-1} \circ R_{x_\ell}(v^j))}{t^j}.$$

Observons que

$$f_{x_\star}(R_{x_\star}^{-1} \circ R_{x_\ell}(v^j)) = f \circ R_{x_\star}(R_{x_\star}^{-1} \circ R_{x_\ell}(v^j)) = f \circ R_{x_\ell}(v^j) = f_{x_\ell}(v^j).$$

On obtient ainsi

$$\begin{aligned} f_{x_\star}^\circ(0; \hat{w}) &= \limsup_{\substack{v^j \rightarrow v_\ell^* \\ t^j \searrow 0}} \frac{f_{x_\star}(R_{x_\star}^{-1} \circ R_{x_\ell}(v^j) + t^j \hat{w}) - f_{x_\ell}(v^j)}{t^j} \\ &\geq \limsup_{k \in K} \frac{f_{x_\star}(R_{x_\star}^{-1} \circ R_{x_\ell}(v^k) + \delta^k \hat{w}) - f_{x_\ell}(v^k)}{\delta^k} \end{aligned}$$

puisque

$$\lim_{k \in K} v^k = v_\ell^*, \quad \lim_{k \in K} d^k = d^* \quad \text{et} \quad \lim_{k \in K} \delta^k = 0.$$

Posons $\xi^k \triangleq R_{x_\star}^{-1} \circ R_{x_\ell}(v^k)$. Les ensembles de sous-niveau de f étant bornés, la suite $\{v^k\}_{k \in K}$ est contenue dans un ensemble compact de $T_{x_\ell} \mathcal{M}$. Puisque $R_{x_\star}^{-1} \circ R_{x_\ell}$ est continue, la suite

$\{\xi^k\}_{k \in K}$ est également contenue dans un compact de $T_{x_*}\mathcal{M}$. L'application R_{x_*} étant $\mathcal{C}^1$, elle est lipschitzienne au voisinage de tout ξ^k . Puisque ces points sont contenus dans un compact, il existe une constante $L_* \in]0, \infty[$ telle que R_{x_*} est L_* -lipschitzienne au voisinage de ξ^k , pour tout $k \in K$. Si f est L_f -lipschitzienne au voisinage de x_* , la composition f_{x_*} est L -lipschitzienne au voisinage de 0 avec $L = L_*L_f$.

Soit $K' \subseteq K$ un ensemble d'indices tel que

$$\lim_{k \in K'} w^k = \hat{w}.$$

On a finalement :

$$\begin{aligned} f_{x_*}^\circ(0; \hat{w}) &\geq \limsup_{k \in K'} \frac{f_{x_*}(\xi^k + \delta^k \hat{w}) - f_{x_*}(\xi^k + \delta^k w^k)}{\delta^k} + \frac{f_{x_*}(\xi^k + \delta^k w^k) - f_{x_\ell}(v^k)}{\delta^k} \\ &\geq \limsup_{k \in K'} -L \|\hat{w} - w^k\| + \frac{f_{x_*}(\xi^k + \delta^k w^k) - f_{x_\ell}(v^k)}{\delta^k} \\ &= \limsup_{k \in K'} \frac{f_{x_*}(\xi^k + \delta^k w^k) - f_{x_\ell}(v^k)}{\delta^k} \\ &= \limsup_{k \in K'} \frac{f_{x_\ell}(v^k + \delta^k d^k) - f_{x_\ell}(v^k)}{\delta^k} \geq 0, \end{aligned}$$

par le lemme 4.2. □

Ce résultat sur $T_{x_*}\mathcal{M}$ est intéressant à plusieurs égards. D'une part, $T_{x_*}\mathcal{M}$ coïncide avec l'ensemble des directions réalisables linéarisées pour les contraintes $h(x) = 0$, objet souvent étudié en optimisation sous contraintes connues. D'autre part, ce résultat ne porte pas sur les dérivées de Clarke de f , mais sur celles de f_{x_*} . Il est donc en un sens plus fort : il indique que lorsqu'on se déplace à partir de 0 selon certaines directions de $T_x\mathcal{M}$, la dérivée directionnelle généralisée reste positive même après rétraction, c'est-à-dire même en suivant la *courbure locale* de $\mathcal{M}$.

Nous terminons en montrant que les directions $\hat{w}$ auxquelles nous nous intéressons existent toujours.

Proposition 4.5. *Avec les objets définis au théorème 4.4, la suite $\{w^k\}_{k \in K}$ admet au moins un point d'accumulation.*

Démonstration. Puisque $R_{x_*}^{-1} \circ R_{x_\ell} : T_{x_\ell}\mathcal{M} \rightarrow T_{x_*}\mathcal{M}$, il existe $\varphi_{(1)}, \varphi_{(2)}, \dots, \varphi_{(n)}$ des fonctions

de $\mathbb{R}^n$ dans $\mathbb{R}$ telles que

$$\forall v \in T_{x_\ell} \mathcal{M}, \quad R_{x_\star}^{-1} \circ R_{x_\ell}(v) = (\varphi_{(1)}(v), \varphi_{(2)}(v), \dots, \varphi_{(n)}(v)).$$

De plus, $R_{x_\star}^{-1} \circ R_{x_\ell}$ étant localement lipschitzienne en tout $v \in T_{x_\ell} \mathcal{M}$, il en va de même pour les $\varphi_{(1)}, \varphi_{(2)}, \dots, \varphi_{(n)}$.

Soient $k \in K$ et $j \in \llbracket 1, n \rrbracket$. Notons $w_{(j)}^k \in \mathbb{R}$ la $j^{\text{ième}}$ composante du vecteur w^k , on a

$$w_{(j)}^k = \frac{\varphi_{(j)}(v^k + \delta^k d^k) - \varphi_{(j)}(v^k)}{\delta^k}.$$

Puisque

$$\lim_{k \in K} v^k = v_\ell^*, \quad \lim_{k \in K} d^k = d^* \quad \text{et} \quad \lim_{k \in K} \delta^k = 0,$$

on en tire

$$\limsup_{k \in K} w_{(j)}^k \leq \varphi_{(j)}^\circ(v_\ell^*; d^*) \leq L_{(j)} \|d^*\|$$

où $L_{(j)}$ est la constante de Lipschitz de $\varphi_{(j)}$. D'autre part, le même raisonnement mène à :

$$\liminf_{k \in K} w_{(j)}^k = -\limsup_{k \in K} -w_{(j)}^k \geq -K_{(j)} \|d^*\|,$$

puisque les fonctions $\varphi_{(j)}$ et $-\varphi_{(j)}$ partagent les mêmes constantes de Lipschitz.

Les limites supérieure et inférieure de chacune des composantes de w^k étant finies, on déduit que $\{w^k\}_{k \in K}$ est bornée, et admet donc au moins un point d'accumulation. $\square$

Les propriétés de convergence dont nous avons su munir RDFO étant énoncées, il est temps de s'intéresser à ce que donne cette méthode lorsqu'elle est mise en œuvre sur différentes instances du problème (P).

CHAPITRE 5 IMPLÉMENTATION ET RÉSULTATS NUMÉRIQUES

Ce chapitre décrit l’implémentation de l’algorithme `RDFO` présenté au chapitre 4, et le teste sur la résolution de problèmes du type de (P). Rappelons que cet algorithme s’insère dans le cadre où les contraintes d’égalité de (P) sont connues, et qu’en pratique, il est nécessaire que les jacobienne et hessienne de h puissent être calculées en tout x satisfaisant $h(x) = 0$. La construction de `RDFO` exige théoriquement que Ω_+ soit une sous-variété de $\mathbb{R}^n$, mais nous montrons que l’implémentation proposée peut donner des résultats pour certains cas dans lesquels cette hypothèse n’est pas satisfaite.

Ce chapitre précise comment la théorie développée au chapitre précédent est mise en œuvre, en se reposant sur le langage `Julia`. L’implémentation proposée repose sur plusieurs librairies importantes pour le calcul numérique sur variétés en `Julia` :  `ManifoldsBase.jl` et  `Manifolds.jl` [19] fournissent les structures fondamentales (variétés, rétractions,...), et  `Manopt.jl` [21] sert de référence pour l’interface conçue, ainsi que pour certains tests.

Un jeu de problèmes test est construit, impliquant des contraintes d’égalité compatibles avec `RDFO`. Ce jeu de problèmes est utilisé pour illustrer le fonctionnement de l’algorithme et l’impact du choix d’une rétraction, ainsi que d’une borne sur son rayon d’inversibilité. Des paramètres optimaux pour le jeu de problèmes construits sont identifiés et avec ceux-ci, `RDFO` est comparé à d’autres solveurs sans dérivées capables de résoudre des instances de (P).

Ce chapitre reprend les notations définies aux chapitres 3 et 4. Notamment, on utilise indifféremment les notations Ω_+ et $\mathcal{M}$ pour désigner le domaine réalisable pour les contraintes d’égalité de (P).

Note au lecteur : Dans ce chapitre, toutes les références à des éléments logiciels présentées en caractères **machine à écrire** sont des liens cliquables vers les librairies, fonctions ou classes mentionnées.

5.1 Implémentation

Cette section décrit les ajustements nécessaires à implémenter la méthode décrite au chapitre 4, puis les choix d’implémentation faits, notamment quant aux rétractions et à leurs rayons d’inversibilité.

5.1.1 Reformulation des sous-problèmes pour l'implémentation

Dans la section 4.1.1, nous prétendions que le sous-problème (4.1) était de dimension q . Il s'agissait là d'une simplification : $T_x\mathcal{M}$ est *isomorphe* à un espace de dimension q mais présenté tel quel, ce problème ne peut être implémenté comme un problème de dimension q . L'implémentation nécessite de décrire la correspondance entre $T_x\mathcal{M}$ et $\mathbb{R}^q$.

Définition 5.1 (Plongement). *Soient $x \in \mathcal{M}$ et $\mathcal{B} = (b_1, b_2, \dots, b_q) \subset \mathbb{R}^n$ une base orthonormée de $T_x\mathcal{M}$. Si $B \triangleq \begin{bmatrix} b_1 & b_2 & \dots & b_q \end{bmatrix} \in \mathbb{R}^{n \times q}$, l'application linéaire*

$$\begin{aligned} \pi_{\mathcal{B}} : \mathbb{R}^q &\rightarrow T_x\mathcal{M} \\ v &\mapsto Bv \end{aligned}$$

est appelée le plongement de $\mathbb{R}^q$ vers $T_x\mathcal{M}$ associé à $\mathcal{B}$.

La proposition 3.2 permet d'affirmer que tout plongement est bijectif. De plus, imposer que la base utilisée soit orthonormée permet d'en faire une isométrie, et ainsi de transférer à $\mathbb{R}^q$ tous les arguments topologiques utilisés dans les propositions et théorèmes du chapitre 4.

Nous nous autorisons un dernier détour théorique : si on pose

$$\begin{aligned} \psi_x : \mathbb{R}^q &\rightarrow \mathcal{M} \\ v &\mapsto R_x \circ \pi_{\mathcal{B}}(v), \end{aligned} \tag{5.1}$$

et que l'on restreint ψ_x au rayon d'inversibilité de R_x , l'application

$$\begin{aligned} \tilde{\psi}_x : B(0, \text{inv}(R_x)) \subseteq \mathbb{R}^q &\rightarrow \psi_x(B(0, \text{inv}(R_x))) \\ v &\mapsto \psi_x(v) \end{aligned}$$

est inversible puisque l'isométrie $\pi_{\mathcal{B}}$ préserve le rayon d'inversibilité. Cette application rejoint un concept central en géométrie différentielle : $\tilde{\psi}_x^{-1}$ est une *carte* sur $\mathcal{M}$. Ces applications sont au centre de la définition traditionnelle qui est donnée d'une variété [22, Définition 8.1].

En pratique, le sous-problème (SP_{ℓ}) est implémenté avec $f_{x_{\ell}} = f \circ \psi_{x_{\ell}}$ et $g_{x_{\ell}} = g \circ \psi_{x_{\ell}}$ et $v \in \mathbb{R}^q$. Ceci nécessite le calcul d'une base orthonormée de $T_{x_{\ell}}\mathcal{M}$.

5.1.2 Bases des espaces tangents

En évacuant le cas trivial où $\mathcal{M} = \Omega_{=}$ est ouvert, le théorème 3.1 permet d'obtenir une base de $T_x\mathcal{M}$ comme une base de $\text{Ker}(J_h(x))$.

Dans l'implémentation proposée, la jacobienne $J_h(x)$ est calculée par différentiation automatique avant à l'aide de la librairie `ForwardDiff.jl` [1]. Une base de $\text{Ker}(J_h(x))$ est ensuite obtenue en effectuant une décomposition en valeurs singulières de $J_h(x)$ et en retenant les vecteurs singuliers de norme inférieure à $q \times \varepsilon$ où $q = \dim \mathcal{M}$ et ε est la précision machine, ce qui est fait grâce au module `LinearAlgebra` de la librairie standard [3].

5.1.3 Rétractions et rayons d'inversibilité

Dans le cas où l'application exponentielle (voir la définition 3.5) est disponible, elle est utilisée comme rétraction. Le plus souvent, le rayon d'injectivité de $\mathcal{M}$ est alors connu, et la borne choisie est $\rho(R_x) = \text{inj}(x)$.

Dans les tests qui suivent, on ne fait appel à l'application exponentielle que pour les problèmes définis sur $\mathbb{S}^{n-1}$. Le rayon d'injectivité est alors constant : pour tout $x \in \mathbb{S}^{n-1}$, $\text{inj}(x) = \pi$. Dans les autres cas, on fait appel à la projection métrique. La rétraction d'un vecteur $v \in T_x \mathcal{M}$ est alors obtenue en résolvant le problème (3.3), dans lequel l'objectif est remplacé par $\frac{1}{2} \|y - (x + v)\|^2$. Ce problème est résolu grâce au solveur `Ipopt` [2] interfacé avec la librairie  `Julia.jl` [50]. La proposition 3.3 affirme que la projection vérifie localement les propriétés d'une rétraction, ce qui permet de conserver les garanties de convergence de `RDFO`, dans le cas où ses autres hypothèses sont satisfaites.

Puisque la projection métrique est localement une rétraction, la proposition 4.1 garantit que son rayon d'inversibilité est strictement positif. Le déterminer est en revanche très difficile, et en obtenir une borne inférieure l'est tout autant. Il existe des travaux fournissant des bornes globales pour tout $x \in \mathcal{M}$ à partir des gradients $\nabla h_i(x)$ ($i \in \llbracket 1, p \rrbracket$) où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ est une fonction localement définissante pour $\mathcal{M}$ [59], mais les calculer requiert un effort de calcul conséquent. Nous proposons de recourir à des quantités artificielles dont nous savons qu'elles ne vérifient pas systématiquement $\rho(R_x) \leq \text{inv}(R_x)$, mais qui sont construites selon l'intuition suivante : le rayon d'inversibilité de la projection métrique est d'autant plus faible que $\mathcal{M}$ est courbée. Afin de recourir à un minimum d'outils, nous ne cherchons pas à calculer la courbure de $\mathcal{M}$, mais postulons qu'elle est proportionnelle aux rayons spectraux des hessiennes $H_{h_i}(x)$. Pour une matrice carrée A de spectre $\text{Sp}(A)$, nous notons

$$\lambda(A) \triangleq \max\{|\alpha| : \alpha \in \text{Sp}(A)\}$$

le rayon spectral de A et

$$\Lambda(h, x) \triangleq \max\{\lambda(H_{h_i}(x)) : i \in \llbracket 1, p \rrbracket\}$$

où H_{h_i} est la matrice hessienne de la $i^{\text{ième}}$ contrainte d'égalité. À partir de ceci, nous proposons les quatre formules suivantes pour le rayon d'inversibilité dans le cas où $R_x(v) = \text{proj}_{\mathcal{M}}(x+v)$:

$$\rho_1(R_x) \triangleq \frac{1}{\Lambda(h, x)}, \quad \rho_2(R_x) \triangleq \frac{1}{\sqrt{\Lambda(h, x)}}, \quad \rho_3(R_x) \triangleq \frac{n}{\Lambda(h, x)}, \quad \rho_4(R_x) \triangleq \frac{n}{\sqrt{\Lambda(h, x)}}. \quad (5.2)$$

Un des buts des sections suivantes est de tester le comportement numérique de RDFO avec chacune de ces bornes.

5.1.4 Traitement des contraintes d'inégalité

Les contraintes d'inégalité sont traitées avec la barrière extrême définie à la section 2.1.3, de sorte que pour tout $v \in T_x \mathcal{M}$,

$$f_x(v) = \begin{cases} f \circ R_x(v) & \text{si } R_x(v) \in \Omega_{\leq} \\ \infty & \text{sinon.} \end{cases}$$

En pratique, on considère que $x \in \Omega_{\leq}$ dès que $\max\{g_j(x) : j \in \llbracket 1, m \rrbracket\} \leq 10^{-8}$. Les bornes sont également traitées de cette façon.

La section suivante illustre numériquement le fonctionnement de RDFO sur un problème de taille 2.

5.2 Première illustration

Nous proposons une première étude du comportement de RDFO en utilisant le problème simple

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & x_1 + 2x_2 \\ \text{s.c.} \quad & x_1^2 + x_2^2 = 1, \end{aligned} \quad (5.3)$$

pour lequel $\Omega_{\leq} = \mathbb{S}^1$. Le solveur sans dérivées utilisé est la version de MADS implémentée dans le logiciel **NOMAD** [17]. Nous utilisons l'interface proposée par la librairie **NOMAD.jl** [5], avec ses paramètres par défaut. La projection métrique est utilisée comme rétraction, munie de la borne ρ_1 définie à l'équation (5.2) pour le rayon d'inversibilité.

En choisissant la solution initiale $x_0 = (1, 0)$, ce problème est résolu par RDFO en quatre itérations externes. Ces itérations sont représentées à la figure 5.1. Pour chacune des itérations, nous représentons l'itéré externe x_ℓ et l'espace tangent $T_{x_\ell} \mathbb{S}^1$. Sur cet espace tangent sont représentés les points $x_\ell + \pi_{\mathcal{B}}(v_\ell^k)$ où les v_ℓ^k sont les points évalués par MADS lors de

la résolution de (SP_ℓ) . En trait pointillé, ces points sont reliés à leurs rétractions sur $\mathbb{S}^1$. La zone d'inversibilité de R_{x_ℓ} est représentée dans $T_{x_\ell}\mathbb{S}^1$ par des parenthèses. Enfin, quelques lignes de niveau de l'objectif sont tracées en arrière-plan.

La figure 5.1(a) montre que le premier sous-problème s'est terminé après une évaluation : le point choisi par MADS a amélioré la valeur de l'objectif mais se situait en dehors du rayon d'inversibilité. La rétraction de ce point devient l'itéré externe x_2 . Le même phénomène se répète dans les figures 5.1(b) et 5.1(c) pour les deuxième et troisième sous-problèmes, ce qui permet à la méthode de se déplacer rapidement dans le quadrant où se situe la solution. La figure 5.1(d) illustre le cas où certains points évalués par MADS se situent en dehors du rayon d'inversibilité, mais où ces points n'améliorent pas la valeur de l'objectif. La résolution du quatrième sous-problème n'est donc pas interrompue, ce qui permet finalement à MADS de converger à l'intérieur du rayon. La rétraction de la solution obtenue dans $T_{x_3}\mathbb{S}^1$ donne la solution finale x_* . Notons que durant cette dernière résolution, l'algorithme a évalué un point dont la rétraction correspond à x_2 , c'est pourquoi le point mauve représentant x_2 est recouvert d'un point bleu. Il s'agit là d'une limitation claire de l'algorithme RDFO tel que présenté ici : en résolvant un nouveau sous-problème, toute l'information acquise lors des sous-problèmes précédents est perdue, ce qui expose au risque de réévaluer des points déjà visités.

Cette illustration en dimension deux sera de nouveau utilisée à la plus tard afin de visualiser les effets sur l'algorithme du choix d'une rétraction et d'une borne ρ associée.

Les trois sections qui suivent proposent des tests numériques sur des ensembles de problèmes de natures différentes : des problèmes dont les contraintes d'égalité sont linéaires, d'autres définis sur des sphères, et enfin des problèmes dont les contraintes d'égalité sont plus variées. Cette séparation permet d'illustrer différents aspects et différentes capacités de RDFO. Elle permet également de le comparer à plusieurs méthodes du chapitre 2 et de la section 3.4 sur des problèmes auxquels elles sont applicables. Nous commençons par la famille de problèmes la plus simple à traiter avec RDFO : les problèmes à contraintes d'égalité linéaires.

5.3 Contraintes d'égalité linéaires

En présence de contraintes d'égalité exclusivement linéaires, l'ensemble $\mathcal{M} = \Omega_-$ correspond à l'ensemble des solutions d'un système d'équations linéaires, donc à un sous-espace affine de $\mathbb{R}^n$. Pour tout $x \in \mathcal{M}$, on a donc $T_x\mathcal{M} = \{-x\} + \mathcal{M}$: les espaces tangents à $\mathcal{M}$ sont de simples translations de $\mathcal{M}$ qui le transforment en un sous-espace vectoriel de $\mathbb{R}^n$. De plus, pour tout $v \in T_x\mathcal{M}$, $\text{proj}_{\mathcal{M}}(x+v) = \text{Exp}_x(v) = x+v$. Enfin, $\text{inv}(\text{proj}_{\mathcal{M}}(x+\cdot)) = \text{inj}(x) = \infty$, et cette valeur est bien obtenue par n'importe laquelle des bornes proposées en (5.2), avec la

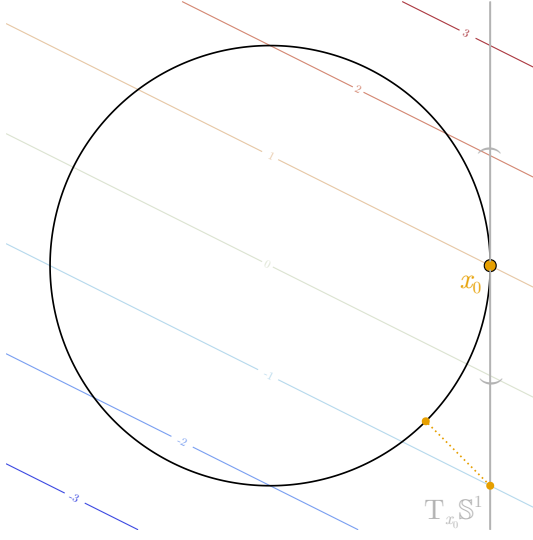
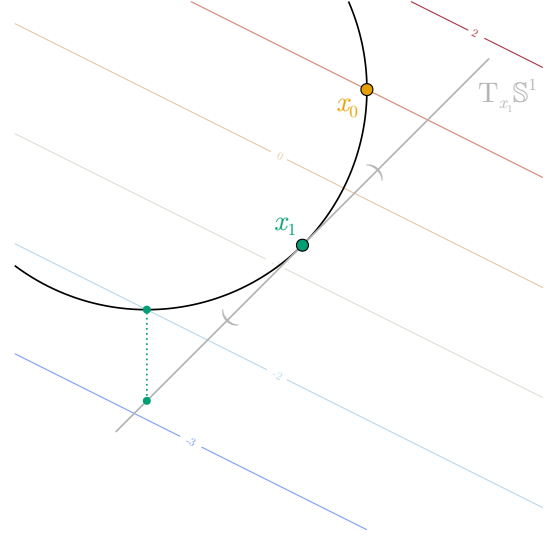
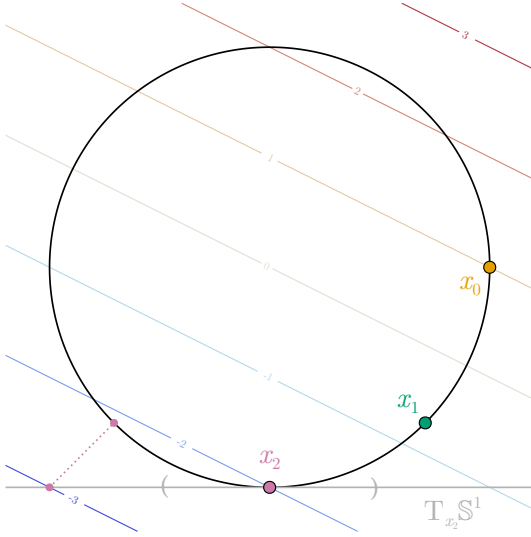
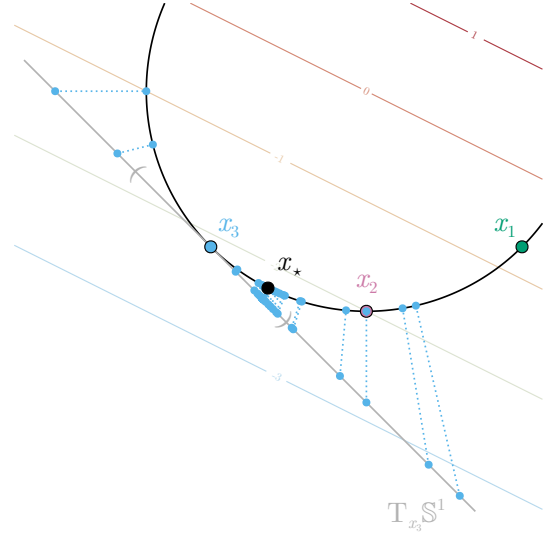
(a) Optimisation dans $T_{x_0}\mathbb{S}^1$.(b) Optimisation dans $T_{x_1}\mathbb{S}^1$.(c) Optimisation dans $T_{x_2}\mathbb{S}^1$.(d) Optimisation dans $T_{x_3}\mathbb{S}^1$ et convergence.

FIGURE 5.1 Illustration du comportement de RDFO sur le problème (5.3).

convention $1/0 = \infty$.

5.3.1 Ensemble de problèmes à contraintes d'égalités linéaires

Les tests sont effectués sur un ensemble de 42 instances pour lesquelles $n \in \llbracket 2, 20 \rrbracket$. Plusieurs instances sont construites avec pour fonction objectif la fonction de Rosenbrock définie pour tout $x \in \mathbb{R}^n$ par

$$f(x) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2, \quad (5.4)$$

et l'ajout de la contrainte

$$\sum_{i=1}^n x_i = 1.$$

Des contraintes de bornes sont également intégrées dans certaines instances. Les autres instances sont construites à partir de la collection de problèmes [43]. Certaines sont adaptées pour ne conserver que les contraintes linéaires. Enfin, les instances sont diversifiées en générant aléatoirement des points de départ différents. Le détail des caractéristiques de ce jeu de problèmes est donné au tableau 5.1, dans lequel la mention « HS » désigne les problèmes issus de la collection [43].

TABEAU 5.1 Constitution du jeu de problèmes test pour les contraintes d'égalité linéaires.

Fonction objectif	n	m (dont bornes)	p	# instances
Rosenbrock	{2, 3, 4, 7, 10, 12, 15, 20}	{0, 1, 2}	1	14
HS48	5	0	1	7
HS49	5	0	2	5
HS51	5	0	3	4
HS52	5	0	3	4
HS53	5	10	3	4
HS112	10	10	3	2
HS119	16	32	8	2
Total				42

5.3.2 Comparaison à la méthode des convertisseurs linéaires

Les problèmes décrits à la section précédente sont résolus avec `RDFO` et avec la méthode des convertisseurs linéaires présentée à la section 2.3.2. L'implémentation de cette méthode offerte par `NOMAD.jl` est utilisée, avec pour convertisseur linéaire la décomposition en valeurs singulières de A . Puisque la méthode des convertisseurs linéaires nécessite un point de départ réalisable pour les contraintes d'égalité, le point de départ fourni si x_0 n'est pas réalisable

est $\text{proj}_{\Omega_{=}}(x_0)$, coïncidant ainsi avec la première étape de RDFO. Les résultats obtenus sont étudiés avec les profils de données définis à la section 2.4. Rappelons que ces deux méthodes sont $\Omega_{=}$ -réalisables : toutes les évaluations qu’elles réalisent sont effectués uniquement en des points respectant les contraintes $h(x) = 0$. Ceci permet de tracer des profils sans se soucier des contraintes d’égalité. Les contraintes d’inégalité sont traitées via la barrière extrême dans les deux cas.

La figure 5.2 présente les profils de données obtenus. Ils révèlent que sur ce jeu de problèmes, la méthode des convertisseurs linéaires est capable dans la majorité des cas d’identifier une solution aussi bonne ou meilleure que RDFO. Ces solutions semblent également pouvoir être parfois obtenues plus rapidement par la méthode des convertisseurs. Cependant, cette différence est minime : quelle que soit la tolérance τ choisie sur le test de convergence, les deux profils obtenus sont très proches. Ceci n’est pas surprenant dans la mesure où ces deux approches sont en réalité presque équivalentes. En effet, si les contraintes d’égalité d’un problème du type de (P) sont toutes des fonctions linéaires, il existe une matrice $A \in \mathbb{R}^{p \times n}$ et un vecteur $b \in \mathbb{R}^p$ tels que $h(x) = Ax - b$, et donc $J_h(x) = A$. Rappelons que l’implémentation proposée ici de RDFO calcule une base de $T_x\mathcal{M}$ par décomposition en valeurs singulières de $J_h(x)$. C’est cette même décomposition qu’utilise la méthode des convertisseurs linéaires. Les solutions qu’elle génère coïncident donc théoriquement de près avec celles proposées par RDFO. Il est possible que les différences observées entre ces deux méthodes soient dues à leurs implémentations.

Les tests menés dans cette section laissent penser que RDFO est une approche viable pour généraliser la méthode des convertisseurs linéaires à des contraintes non linéaires, en perdant peu de la performance qu’elle propose. La deuxième famille de problèmes qui nous intéresse est celle des problèmes définis sur des sphères.

5.4 Problèmes définis sur des sphères

La méthode RDFO est présentée dans ce mémoire comme une méthode conçue pour résoudre des problèmes du type de (P) lorsque $\Omega_{=}$ constitue une sous-variété riemannienne de $\mathbb{R}^n$. Si nous mettons de côté la définition de $\mathcal{M}$ comme l’ensemble réalisable pour les contraintes $h(x) = 0$, RDFO est en réalité une méthode capable de résoudre des problèmes d’optimisation sans dérivées définis sur des variétés riemanniennes. Ces deux approches sont équivalentes, mais n’incitent pas à s’intéresser à la même littérature.

La littérature de l’optimisation sur variétés étudie des problèmes dans lesquels la structure de variété apparaît naturellement, et beaucoup moins ceux dans lesquels elle est « imposée »

ou supposée, comme c'est le cas dans l'approche de RDFO présentée ici. Cependant, ces problèmes sont rarement définis sur $\mathbb{R}^n$. Ils concernent bien souvent des variétés matricielles comme les variétés de Stiefel

$$\text{St}(p, n) = \{X \in \mathbb{R}^{n \times p} : X^\top X = I_p\}$$

ou celles des matrices de rang $r \geq 0$

$$\mathbb{R}_r^{m \times n} = \{X \in \mathbb{R}^{m \times n} : \text{rang}(X) = r\},$$

ou encore les grassmanniennes $\text{Gr}(n, k)$, qui désignent l'ensemble des sous-espaces vectoriels de $\mathbb{R}^n$ de dimension k [7, 22]. Certains de ces problèmes pourraient être résolus avec RDFO (moyennant reformulation), mais le cadre de ce mémoire étant spécifiquement l'utilisation de l'optimisation riemannienne pour la résolution de problèmes sous contraintes d'égalité, nous les laissons de côté. La littérature de l'optimisation riemannienne étudie cependant un problème spécifique défini sur $\mathbb{S}^{n-1}$: celui du calcul de valeurs propres extrêmes de matrices symétriques.

5.4.1 Problèmes de valeurs propres extrêmes

Si $A \in \mathbb{R}^{n \times n}$ est une matrice symétrique, on peut montrer que le quotient de Rayleigh

$$x \mapsto \frac{\langle x, Ax \rangle}{\langle x, x \rangle}$$

admet pour minimum la plus petite valeur propre de A , et qu'il est atteint en tout vecteur propre associé à cette valeur propre. De la même manière, le maximum de ce quotient est la plus grande valeur propre de A , atteint en tout vecteur propre correspondant [7, Prop. 4.6.1]. Ce quotient étant invariant par mise à l'échelle, on peut se contenter d'étudier les vecteurs unitaires. La plus petite valeur propre de A et un vecteur propre associé peuvent donc être calculés en résolvant le problème

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \langle x, Ax \rangle \\ \text{s.c.} \quad & \|x\| = 1, \end{aligned} \quad \text{i.e.} \quad \min_{x \in \mathbb{S}^{n-1}} \langle x, Ax \rangle. \quad (5.5)$$

Le problème de maximisation fournit la plus grande valeur propre, et un vecteur propre unitaire associé. De plus, dès lors que A admet plusieurs valeurs propres distinctes, le problème (5.5) est symétrique puisque $\langle x, Ax \rangle = \langle -x, -Ax \rangle$, pour tout $x \in \mathbb{R}^n$. La figure

figure 5.3 montre la résolution de ce problème par RDFO avec la matrice

$$A = \begin{bmatrix} 1 & -3 \\ -3 & -2 \end{bmatrix}$$

dont les valeurs propres sont $\lambda_{\min} = \frac{-1-3\sqrt{5}}{2} \approx -3.85$ et $\lambda_{\max} = \frac{-1+3\sqrt{5}}{2} \approx 2.85$. Les espaces tangents ne sont pas représentés cette fois-ci, afin d'alléger la figure. Les courbes de niveau représentées sont celles de $x \mapsto \frac{\langle x, Ax \rangle}{\langle x, x \rangle}$; elles montrent bien que ce quotient est invariant par mise à l'échelle de x , d'où la pertinence de restreindre l'étude à $\mathbb{S}^1$. Sur la figure 5.3(a), ces courbes de niveau rendent claire la symétrie du problème. La figure 5.3(b) représente davantage de courbes de niveau du quotient au voisinage de la solution identifiée par RDFO.

Les problèmes de cette nature sont intéressants, puisqu'il est possible d'en générer des instances diverses *via* des matrices symétriques aléatoires. Nous les utilisons pour créer un jeu de problèmes définis sur des sphères.

5.4.2 Ensemble de problèmes définis sur des sphères

Nous testons RDFO sur un ensemble de 49 instances de problèmes définis sur des sphères, pour lesquelles $n \in \llbracket 2, 30 \rrbracket$. Plusieurs instances sont construites à partir du problème (5.5), en générant aléatoirement des matrices symétriques, en modifiant les points de départ et en étudiant les problèmes de minimisation et de maximisation. Ces instances sont toutes définies sur des sphères unité, puisque leurs objectifs sont invariants par mise à l'échelle. De la même manière que pour construire notre ensemble de problèmes à contraintes d'égalité linéaires, nous y ajoutons des problèmes obtenus à partir de la fonction objectif de Rosenbrock (5.4), ou de fonctions issues de problèmes de la suite [43]. Nous varions également les rayons des sphères et les points de départ. Aucune contrainte d'inégalité n'est intégrée dans cet ensemble de test. Les caractéristiques des instances produites sont résumées au tableau 5.2.

Cet ensemble de test est utilisé pour étudier l'effet qu'a le choix d'une rétraction et d'une borne associée pour le rayon d'inversibilité.

5.4.3 Choix d'une rétraction et d'une borne sur le rayon d'inversibilité

Dans la méthode RDFO telle que présentée jusqu'ici, plusieurs paramètres peuvent être ajustés, le premier d'entre eux étant le solveur sans dérivées utilisé. Nous choisissons ici de mener toutes nos expériences avec MADS pour solveur.

Deux paramètres qui peuvent être déterminants pour les performances de la méthode sont la

TABLEAU 5.2 Constitution du jeu de problèmes test sur des sphères.

Fonction objectif	n	Rayon r	# instances
Valeurs propres extrêmes	$\{2, 3, 5, 7, 10, 15, 30\}$	1	28
Rosenbrock	$\{2, 3, 5, 7, 15\}$	$r \in \{1, 2, 4\}$	8
HS5	2	1	2
HS25	3	$r \in \{50, 100\}$	4
HS54	6	10^8	2
HS62	3	$r \in \{0.75, 1\}$	3
HS110	10	30	2
Total			49

rétraction employée et la borne associée pour le rayon d'inversibilité. Nous illustrons ceci en revenant à la figure 5.1, qui présentait le comportement de RDFO sur le problème (5.3) avec $R_x(v) = \text{proj}_{\mathbb{S}^1}(v)$ et $\rho(R_x) = 1/\Lambda(h, x)$. Dans la figure 5.4, nous représentons la résolution du même problème avec d'autres rétractions et/ou rayons d'inversibilité.

Dans le cas où la projection est utilisée comme rétraction, les quatre bornes sur $\text{inv}(R_x)$ proposées en (5.2) prennent les valeurs suivantes pour tout $x \in \mathbb{S}^1$:

$$\rho_1(R_x) = \frac{1}{2}, \quad \rho_2(R_x) = \frac{\sqrt{2}}{2}, \quad \rho_3(R_x) = 1, \quad \rho_4(R_x) = \sqrt{2}.$$

L'utilisation de ρ_2 et de ρ_3 pour résoudre le problème (5.3) fournit des résultats semblables à ceux obtenus avec ρ_1 : le problème est résolu en trois itérations externes. La borne ρ_4 est en revanche suffisamment permissive quant à l'interruption des sous-problèmes, et mène à la résolution du même problème en deux itérations externes. Ce comportement est illustré à la figure 5.4(a). Lors de la résolution du premier sous-problème, MADS génère le point $v_0^1 = (0, -1) \in T_{x_0}\mathbb{S}^1$, qui vérifie $f(R_{x_0}(v_0^1)) < f(x_0)$. Puisque $\|v_0^1\|_{x_0} \geq \rho_1(R_{x_0})$, la résolution du premier sous-problème se termine lorsque ρ_1 est utilisée. À l'inverse, $\|v_0^1\|_{x_0} < \rho_4(x_0)$. La résolution se poursuit donc. La stratégie opportuniste de MADS génère le point $v_0^2 = (0, -4)$ qui améliore une nouvelle fois l'objectif, mais $\|v_0^2\|_{x_0} \geq \rho_4(R_{x_0})$. Un second sous-problème est résolu à partir de $x_1 \triangleq R_{x_0}(v_0^2)$. L'espace tangent n'est pas représenté pour ne pas surcharger la figure, mais la tolérance liée à la borne ρ_4 permet à l'algorithme de converger dès le second sous-problème. Notons que certaines évaluations ont été menées en dehors du rayon d'inversibilité sur $T_{x_1}\mathbb{S}^1$ mais puisqu'elles n'améliorent pas l'objectif, la résolution n'est pas interrompue.

Puisque le problème que nous étudions est défini sur $\mathbb{S}^1$, il nous donne l'occasion d'observer l'effet de l'utilisation de l'application exponentielle en utilisant sa forme analytique donnée en

(3.5). Son rayon d'inversibilité est de plus connu en tout point : il est appelé le *rayon d'injectivité* de $\mathbb{S}^1$ et vaut π . Avec $x_0 = (1, 0)$, Exp_{x_0} crée une bijection entre $\mathbb{S}^1$ et le segment $] -\pi, \pi]$ de $T_{x_0}\mathcal{M}$. Ceci permet à RDFS d'identifier la solution du problème étudié en une seule itération externe, c'est ce que montre la figure 5.4(b). Il est important de repérer sur cette figure les traits pointillés qui lient les éléments de $T_{x_0}\mathbb{S}^1$ à leurs images par Exp_{x_0} . Le premier point v_0^1 évalué dans le sous-problème est le même que dans la figure 5.4(a) ; on constate en comparant ces deux figures que $\text{Exp}_{x_0}(v_0^1) \neq \text{proj}_{\mathbb{S}^1}(v_0^1)$. L'utilisation de l'application exponentielle ne concerne donc pas que le rayon d'inversibilité. Précisons que tous les points évalués en dehors de la zone d'inversibilité de Exp_{x_0} sont effacés de la figure 5.4(b), afin de la rendre plus lisible. Notamment, la stratégie opportuniste de MADS génère $v_0^2 = (0, -4)$, tout comme elle le fait lorsque ρ_4 est utilisée. Pourtant dans ce cas, malgré le fait que $\|v_0^2\|_{x_0} > \text{inj}(x) = \pi$, la résolution du sous-problème se poursuit. En effet, $\text{Exp}_{x_0}(v_0^2) \approx (-0.654, 0.757)$ se situe dans le dernier quadrant du cercle, contrairement à la projection du même point. La valeur de l'objectif associée n'étant pas inférieure à celle de la solution courante, la résolution n'est pas interrompue.

Ces observations sont tout à fait spécifiques à la sphère et à la fonction objectif choisie. Nous les complétons en résolvant les problèmes de l'ensemble de test décrit à la section précédente avec la projection métrique munie de chacune des quatre bornes proposées, puis avec l'application exponentielle. L'application exponentielle se généralise aux sphères de rayon $r > 0$ par la formule suivante :

$$\text{Exp}_x(v) = \cos\left(\frac{\|v\|}{r}\right)x + r \sin\left(\frac{\|v\|}{r}\right)\frac{v}{\|v\|},$$

et $\text{inj}(x) = \pi r$. Les profils de données obtenus sont représentés à la figure 5.5.

Ces profils révèlent que la projection munie des bornes ρ_1 et ρ_2 , qui ne prennent pas en compte les dimensions des problèmes, fournit des performances généralement moins bonnes que lorsqu'elle est munie des bornes ρ_3 et ρ_4 , qui sont proportionnelles à ces dimensions. Ils révèlent également que l'application exponentielle, dont la figure 5.4(b) laisse penser qu'elle constitue un choix optimal de rétraction, n'est pas toujours à privilégier. Sur le jeu de problèmes utilisé, elle a besoin de plus d'évaluations pour parvenir à des solutions aussi bonnes que la projection munie de la borne ρ_3 . Le profil tracé à la précision $\tau = 10^{-5}$ montre même que la projection munie de ρ_3 est capable d'identifier de meilleures solutions que l'application exponentielle.

Nous proposons une explication potentielle à cette observation. Le rayon d'injectivité (qui quantifie l'inversibilité de Exp) souffre du même problème que les bornes ρ_1 et ρ_2 : il est

indifférent à la dimension de la sphère. Sur des problèmes de grande dimension, MADS ne peut donc générer une suite d'itérés restant à l'intérieur de la zone d'inversibilité que si le paramètre de taille du treillis est réduit très rapidement. L'implémentation effectuée n'encourage pas cela, en imposant au début de chaque sous-problème la valeur de 1 à ce paramètre dans toutes les directions de $T_x\mathcal{M}$.

Parmi les choix de rétraction et de borne ρ envisagés, la projection munie de la borne $\rho_3(R_x) = n/2$ fournit les meilleurs résultats sur l'ensemble test choisi. Nous utilisons ces paramètres pour comparer RDFO à deux autres implémentations de MADS capables de résoudre les mêmes problèmes.

5.4.4 Comparaison à d'autres implémentations de MADS

Les problèmes de notre ensemble de test étant tous définis sur des sphères, ils peuvent être résolus avec deux autres implémentations de MADS. La première est l'implémentation présente dans la librairie 🍌 `Manopt.jl` [4], développée dans le cadre du travail [57] sur le problème de Procrustes cité à la section 3.4. Cette implémentation utilise exclusivement l'application exponentielle comme rétraction, et le rayon d'injectivité de la variété étudiée. Elle a également recours à l'application logarithmique, inverse de l'application exponentielle. L'implémentation repose sur les versions proposées dans 🍌 `Manifolds.jl`. La seconde implémentation que nous étudions utilise le passage en coordonnées sphériques : la sphère de $\mathbb{R}^n$ de rayon $r > 0$ centrée en l'origine est paramétrisée par les angles $(\theta_1, \theta_2, \dots, \theta_{n-1})$ grâce à la transformation

$$\begin{cases} x_i = r \cos(\theta_i) \prod_{j=1}^{i-1} \sin(\theta_j) \text{ pour } i \in \llbracket 1, n-1 \rrbracket \\ x_n = r \prod_{j=1}^{n-1} \sin(\theta_j), \end{cases}$$

où $\theta_1, \theta_2, \dots, \theta_{n-2} \in [0, \pi]$ et $\theta_{n-1} \in [0, 2\pi]$, ce qui permet la même réduction de dimension que les approches par résolution sur les espaces tangents.

Les résultats de ces tests sont présentés à la figure 5.6. Les profils de données peuvent être comparés à ceux de la figure 5.5, qui ont été générés à partir du même jeu de problèmes. On constate ainsi que la proportion des problèmes résolus par RDFO avec projection a diminué d'environ 10%, suggérant que l'introduction des implémentations de 🍌 `Manopt.jl` et par coordonnées sphériques ont apporté de meilleures solutions aux problèmes de l'ensemble de test. Sur cet ensemble, on constate que l'implémentation de MADS présente dans 🍌 `Manopt.jl` est capable de produire des solutions de meilleure qualité que les autres méthodes, mais que plus d'évaluations de l'objectif sont nécessaires pour y parvenir. Il est utile de remarquer que les

profils de données aux précisions $\tau = 10^{-3}$ et $\tau = 10^{-5}$ sont tracés entre 1 et 1000 groupes de $n + 1$ évaluations (contrairement à la précision 10^{-2}), et illustrent bien ce fait. L’approche par coordonnées sphériques, quant à elle, fournit des résultats peu convaincants.

5.5 Contraintes d’égalité non linéaires

Pour terminer, nous testons RDFO sur la catégorie la plus large de problèmes qu’il est capable de résoudre : les problèmes à contraintes d’égalité non linéaires quelconques.

5.5.1 Banque de problèmes

Les tests sont menés sur un ensemble de 63 instances contraintes par des égalités non linéaires quelconques, et de dimensions variant entre 2 et 7. Ces instances sont toutes issues de [43], et utilisées avec des points de départs variés. Le détail de l’ensemble utilisé est reporté au tableau 5.3.

TABLEAU 5.3 Constitution du jeu de problèmes test pour les contraintes d’égalité non linéaires.

Fonction objectif	n	p	# instances
HS6	2	1	5
HS7	2	1	5
HS26	3	1	10
HS27	3	1	7
HS39	4	2	7
HS40	4	3	7
HS42	4	2	7
HS47	5	3	8
HS56	7	4	7
Total			63

5.5.2 Résultats comparatifs

Les problèmes de la section précédente sont résolus avec deux méthodes : RDFO muni de la projection métrique et de la borne ρ_3 , et la méthode de région de confiance COBYLA (voir la section 2.2.2). L’implémentation par défaut de COBYLA proposée dans la librairie PRIMA.jl [6] est utilisée. Les résultats de ces expériences sont proposés à la figure 5.7.

Rappelons que COBYLA n’est pas une méthode réalisable, et qu’elle n’a accès à aucune information concernant les contraintes (d’égalité ou d’inégalité). Il est donc normal d’observer

que RDFO aboutit plus rapidement à une solution, puisqu'elle peut concentrer ses efforts sur l'amélioration de la valeur de l'objectif sans se préoccuper de la faisabilité. Cependant, une fois la faisabilité atteinte par COBYLA, les résultats obtenus sont sensiblement de même qualité que ceux produits par RDFO. COBYLA est même capable de résoudre légèrement plus de problèmes que RDFO à basse précision, mais la tendance s'inverse avec l'augmentation de la précision – les solutions identifiées par RDFO sont donc de meilleure qualité sur une partie des instances étudiées. Des expériences plus poussées seraient nécessaires, notamment sur des jeux de problèmes de plus grande dimension, et incluant plus de contraintes d'inégalité.

Notons que les effets des rétractions et des bornes sur $\text{inv}(\mathbf{R}_x)$ n'ont été évalués que sur des problèmes définis sur des sphères. Il serait pertinent d'effectuer les mêmes tests en se restreignant à la projection, afin d'évaluer l'effet des bornes proposées à l'équation (5.2).

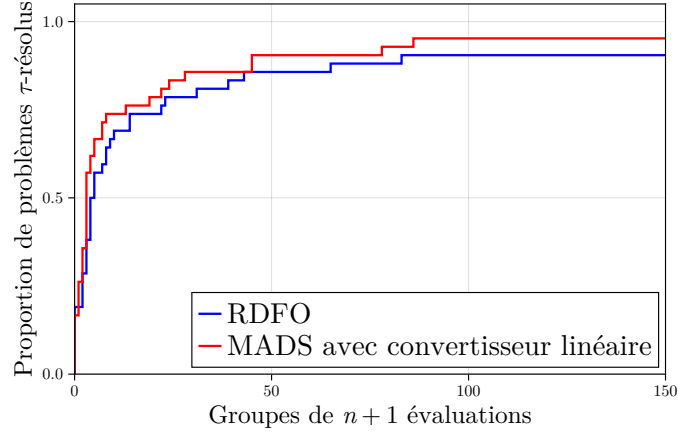
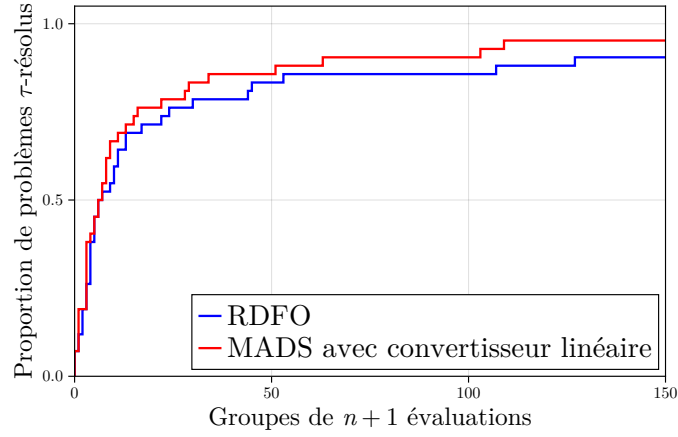
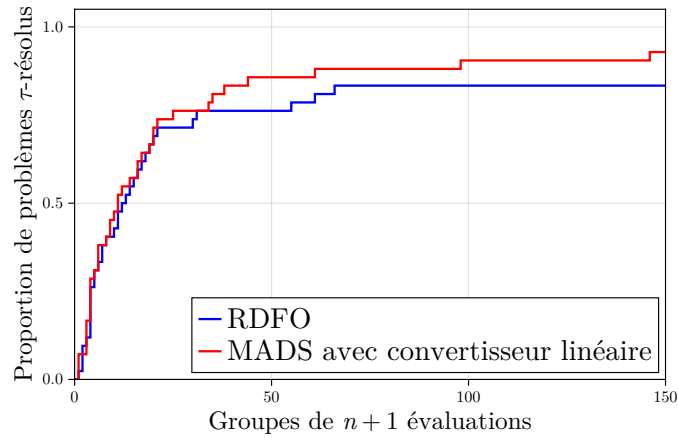
(a) $\tau = 10^{-2}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

FIGURE 5.2 Profils de données sur l'ensemble de problèmes test à contraintes d'égalité linéaires.

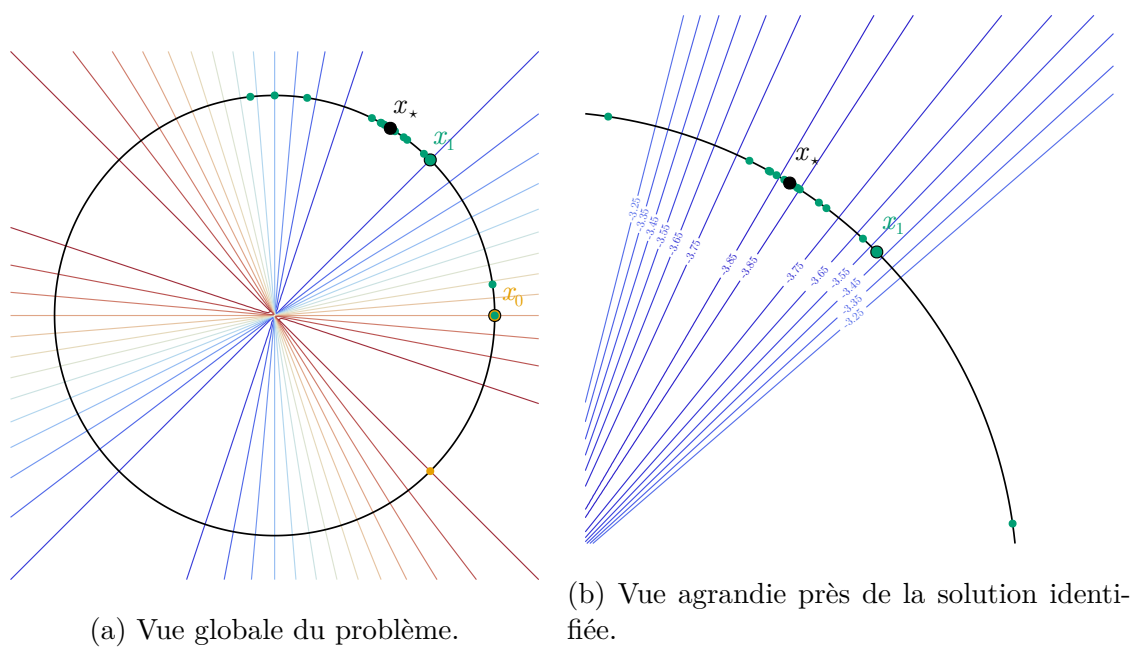
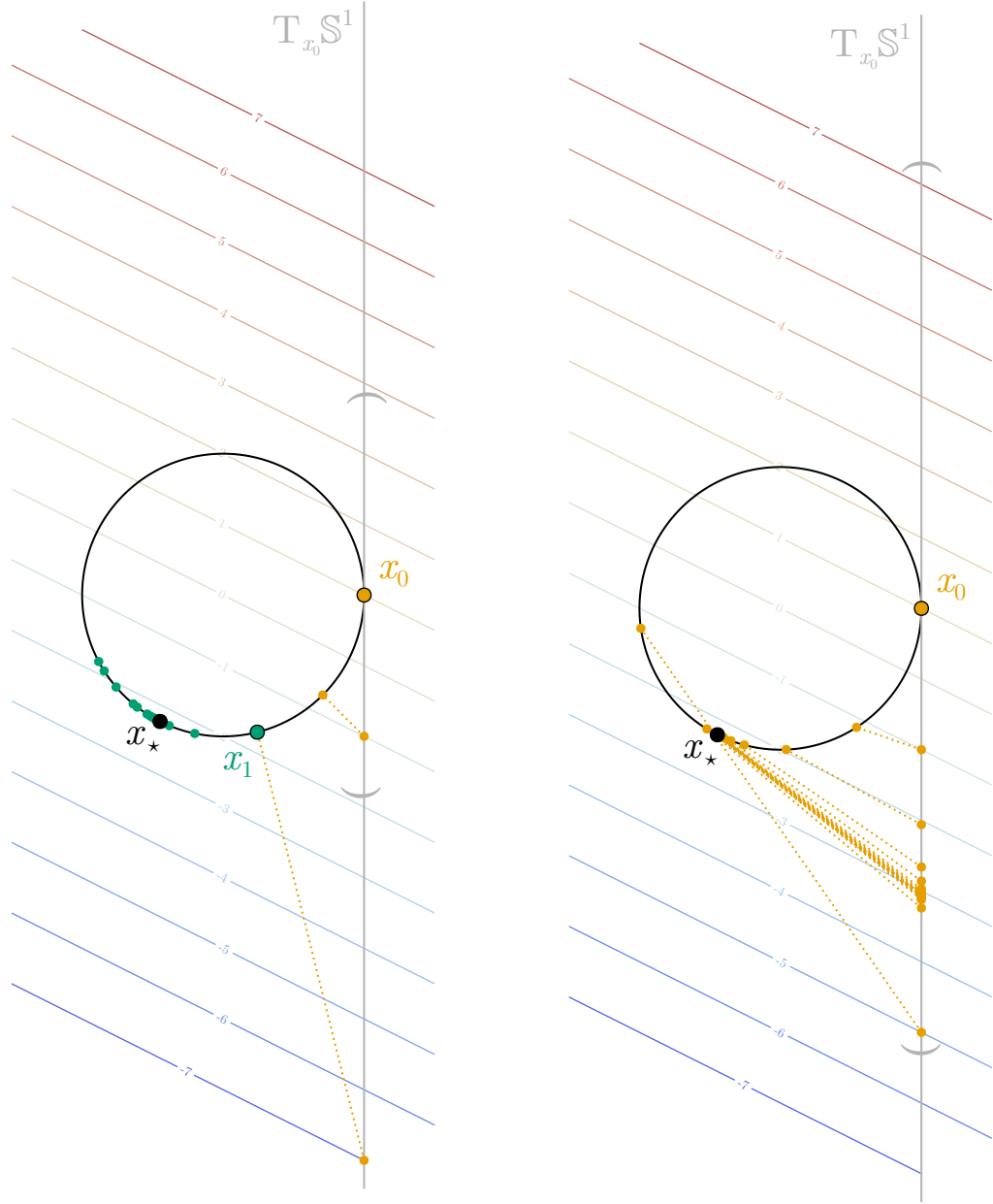


FIGURE 5.3 Résolution du problème (5.5) de plus petite valeur propre par RDFO.



(a) $R_x(v) = \text{proj}_{\mathbb{S}^1}(x + v)$ et $\rho(x) = \frac{2}{\sqrt{\Lambda(h,x)}}$. (b) $R_x(v) = \text{Exp}_x(v)$ et $\rho(x) = \text{inj}(x) = \pi$.

FIGURE 5.4 Comportement de RDOF avec différentes rétractions et rayons d'inversibilité sur le problème (5.3).

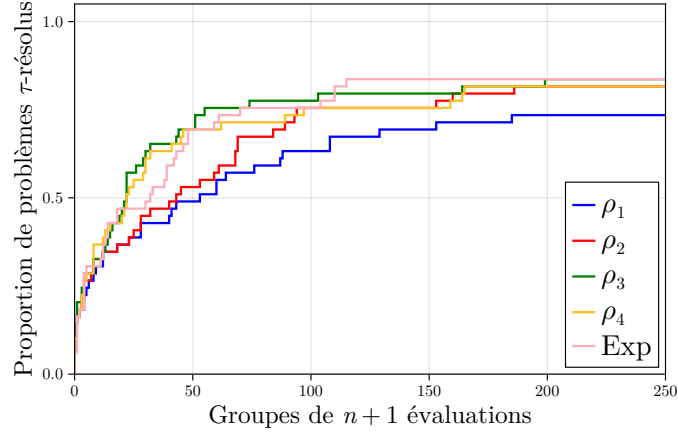
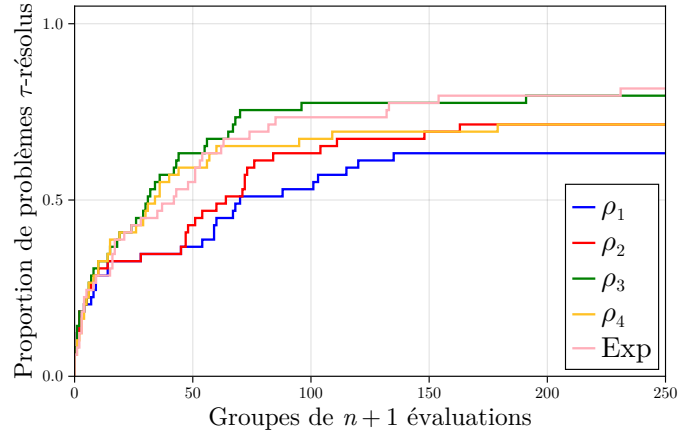
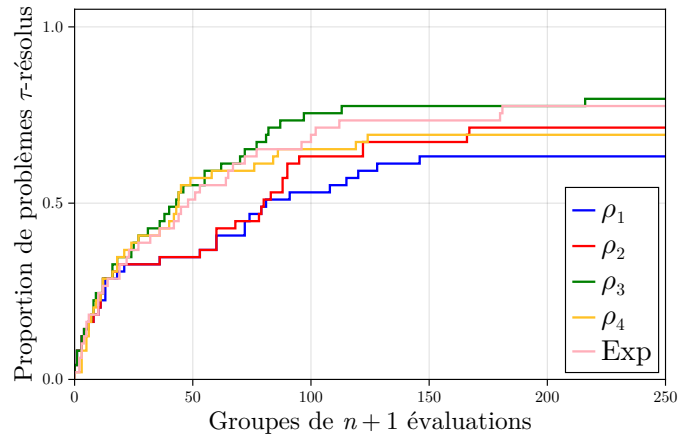
(a) $\tau = 10^{-2}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

FIGURE 5.5 Profils de données comparant les différentes rétractions et les rayons d'inversibilité associés sur l'ensemble de problèmes test définis sur des sphères.

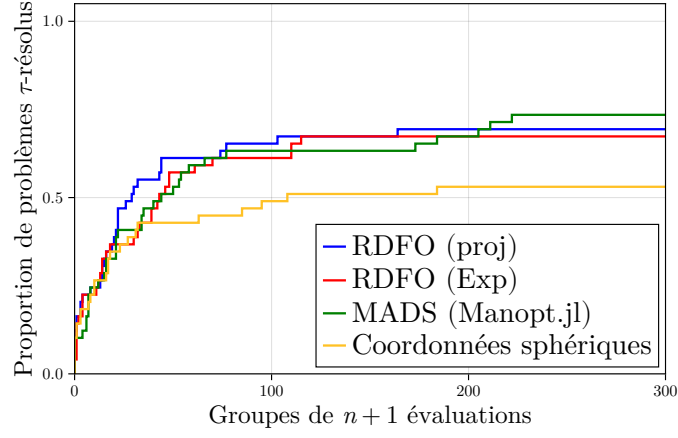
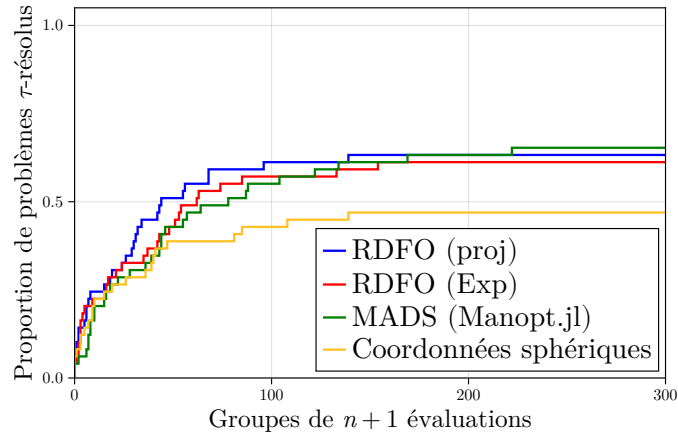
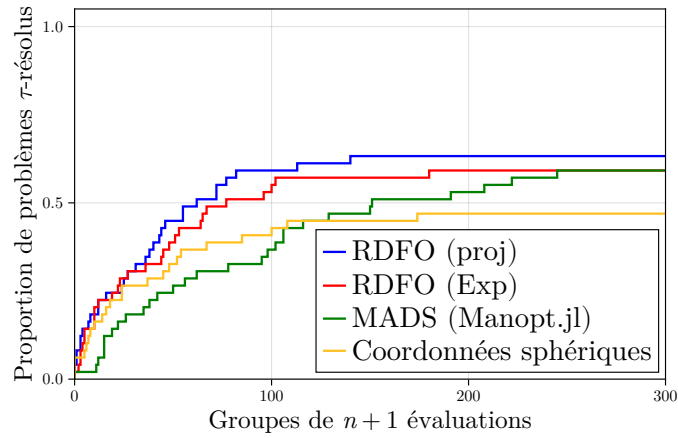
(a) $\tau = 10^{-2}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

FIGURE 5.6 Profils de données sur l'ensemble de problèmes test définis sur des sphères.

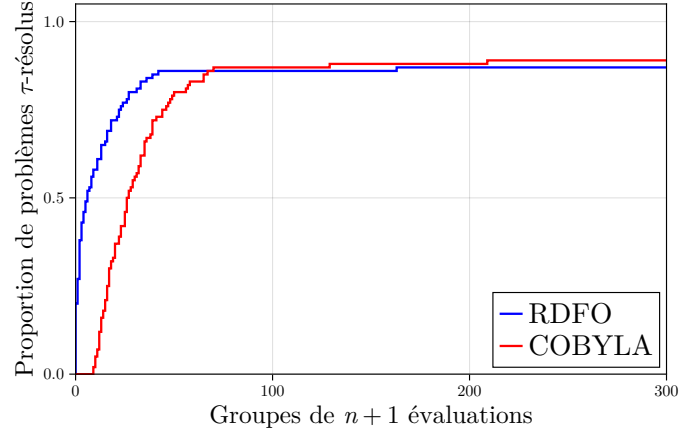
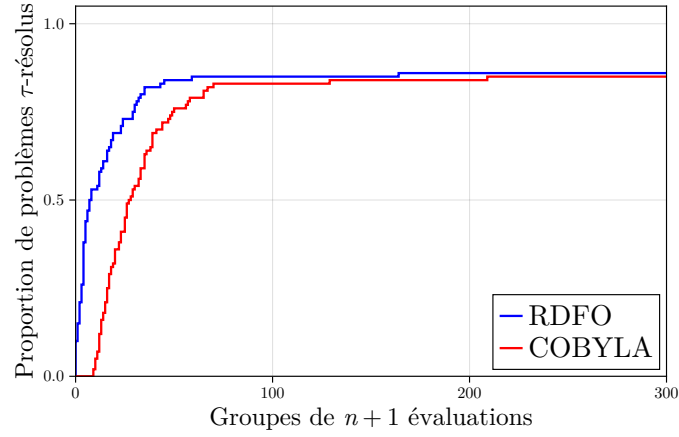
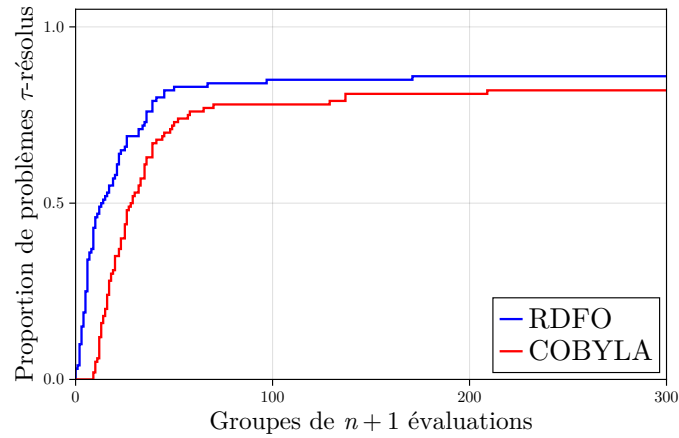
(a) $\tau = 10^{-2}$ (b) $\tau = 10^{-3}$ (c) $\tau = 10^{-5}$

FIGURE 5.7 Profils de données sur l'ensemble de problèmes test à contraintes d'égalité non linéaires.

CHAPITRE 6 CONCLUSION

6.1 Synthèse des contributions

Ce mémoire a abordé la résolution des problèmes d'optimisation de la famille de (P) par des méthodes sans dérivées ; plus précisément, par des méthodes de recherche directe. La démarche suivie prend acte des limites à la fois théoriques et pratiques que pose la présence de contraintes d'égalité pour des méthodes dont la théorie de convergence repose sur la notion de cône hypertangent à un ensemble. Forte de ce constat, cette démarche étudie le cas dans lequel les contraintes d'égalité sont connues, et tente de tirer profit des informations sur la géométrie de l'ensemble $\Omega_=_$ qu'apporte cette connaissance.

Dès lors que les contraintes d'égalité ne sont pas linéaires, l'ensemble $\Omega_=_$ se retrouve à la fois courbe et d'intérieur vide, deux propriétés qui mettent en difficulté les méthodes de recherche directe. Cependant, dans le cas où cet ensemble est suffisamment régulier pour former une sous-variété riemannienne de $\mathbb{R}^n$, la linéarité peut être retrouvée localement par les espaces tangents. La méthode RDFO proposée est fondée sur ce constat (partagé par la littérature) : elle propose de résoudre des sous-problèmes successifs sur les espaces tangents à $\Omega_=_$ à l'aide d'une méthode de recherche directe. Cette approche se comporte comme une façon systématique de paramétrer l'ensemble $\Omega_=_$ par au moins $n - p$ degrés de liberté, où $\Omega_=_ \subseteq \mathbb{R}^n$ et p est le nombre de contraintes d'égalité définissant $\Omega_=_$.

La correspondance locale entre $\Omega_=_$ et ses espaces tangents est contrôlée par la notion de *rayon d'inversibilité* d'une rétraction. Dans le cas où la convergence du solveur utilisé est observée sur un sous-problème à l'intérieur de ce rayon, nous avons montré à l'aide de l'opération de *transfert*, que des conditions d'optimalité pouvaient être transmises d'un espace tangent à l'autre.

La mise en pratique de cette méthode nécessite plusieurs choix empiriques, notamment celui de bornes artificielles sur le rayon d'inversibilité. En proposant plusieurs valeurs pour ces bornes, nous avons mené des observations concernant le comportement de RDFO sur plusieurs instances de problèmes d'optimisation. Les observations menées en faible dimension lorsque MADS est utilisé comme solveur confirment l'intuition selon laquelle la capacité du couple $(R_x, \text{inv}(R_x))$ à décrire la correspondance entre $\Omega_=_$ et son espace tangent en x peut être déterminante pour le comportement de la méthode.

Finalement, les comparatifs menés avec d'autres méthodes de la littérature mènent à la conclusion suivante : RDFO est capable de résoudre des instances de (P) de natures plus

diverses que d'autres solveurs, et en offrant une approche Ω_- -réalisable qui peut s'avérer décisive en pratique. Les solutions que cette méthode est capable de produire sur les ensembles de problèmes considérés sont dans de nombreux cas comparables à celles obtenues par des solveurs dédiés, ou par des méthodes non réalisables, mais rarement meilleures. La vocation de cette implémentation n'est pas de concurrencer des solveurs spécialisés dans la résolution de problèmes spécifiques (à contraintes linéaires ou d'optimisation riemannienne, par exemple), mais d'offrir à la fois une méthode réunissant les capacités de ces solveurs, et ouvrant une nouvelle voie au traitement des contraintes d'égalité non linéaires dans les situations où une méthode Ω_- -réalisable est requise.

6.2 Limitations et perspectives de poursuite des travaux

La méthode et les tests présentés dans ce mémoire sont pensés comme une preuve de concept, qui peut bénéficier de nombre d'améliorations.

Le choix d'un couple $(R_x, \rho(R_x))$ peut être crucial pour les performances de RDFO, particulièrement dans les problèmes de grande dimension. D'une part, les projections ont pour principal défaut un rayon d'inversibilité certes strictement positif, mais relativement peu prédictible. D'autre part, le comportement de l'application exponentielle peut s'avérer peu contrôlable en grande dimension. Nous pensons qu'à court terme, peu de bénéfices seraient tirés d'une réflexion autour de rétractions alternatives. Au contraire, une direction de recherche intéressante consisterait à développer des moyens systématiques d'adapter au couple $(R_x, \rho(R_x))$ choisi les solveurs utilisés pour la résolution des sous-problèmes. À titre d'exemple, l'instabilité relative de l'application exponentielle pourrait être palliée par une mise à l'échelle des variables.

Une autre limitation de la méthode telle qu'elle est proposée ici réside dans la perte d'information que subit le solveur lors du passage d'une itération externe à une autre. L'implémentation de MADS proposée par le logiciel **NOMAD** construit des modèles quadratiques de l'objectif et des contraintes au fil des évaluations qu'elle entreprend. L'utilisation de ces modèles lors des phases de recherche (*search*) de MADS peut rendre la méthode très efficace. Dans le cas où une itération externe $\ell \in \mathbb{N}$ se termine, il serait intéressant de pouvoir transférer à $T_{x_{\ell+1}}\mathcal{M}$ les données des évaluations effectuées dans tous les espaces tangents visités jusqu'en x_ℓ . Ceci pourrait impliquer la notion de transport vectoriel, qu'utilise l'implémentation proposée dans  **Manopt.jl** pour transporter d'un espace tangent à l'autre le treillis de MADS.

De plus, des implémentations d'algorithmes comme MADS sont pensées pour équilibrer les comportements d'exploration et d'exploitation. Arrêter prématurément la résolution d'un

sous-problème dès lors qu'une amélioration est trouvée en dehors de la région d'inversibilité de R_x peut nuire à la façon dont le solveur déploie cet équilibre dans $T_x\mathcal{M}$. De nouveau, permettre au solveur de se déplacer entre les espaces tangents en conservant les informations acquises lors des itérations extérieures précédentes serait un gain précieux.

RÉFÉRENCES

- [1] ForwardDiff.jl. <https://github.com/JuliaDiff/ForwardDiff.jl>. URL <https://github.com/JuliaDiff/ForwardDiff.jl>. (dernière consultation le 27/03/2026).
- [2] Ipopt.jl. <https://github.com/jump-dev/Ipopt.jl>. URL <https://github.com/jump-dev/Ipopt.jl>. (dernière consultation le 27/03/2026).
- [3] Linear Algebra module of the Julia Standard Library. <https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/>. URL <https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/>. (dernière consultation le 27/03/2026).
- [4] Manopt.jl : Mesh Adaptive Direct Search Solver. https://manoptjl.org/stable/solvers/mesh_adaptive_direct_search/. URL https://manoptjl.org/stable/solvers/mesh_adaptive_direct_search/. (dernière consultation le 28/03/2026).
- [5] NOMAD.jl. <https://bbopt.github.io/NOMAD.jl/stable/>. URL <https://bbopt.github.io/NOMAD.jl/stable/>. (dernière consultation le 27/03/2026).
- [6] PRIMA.jl. <https://github.com/libprima/PRIMA.jl>. URL <https://github.com/libprima/PRIMA.jl>. (dernière consultation le 30/03/2026).
- [7] P.-A. Absil, R. Mahony et R. Sepulchre : *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2007. URL <https://press.princeton.edu/absil>.
- [8] P.-A. Absil et J. Malick : Projection-like Retractions on Matrix Manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012. URL <https://doi.org/10.1137/100802529>.
- [9] S. Alarie, C. Audet, A. Gheribi, M. Kokkolaras et S. Le Digabel : Two decades of blackbox optimization applications. *EURO Journal on Computational Optimization*, 9:100011, 2021. URL <https://doi.org/10.1016/j.ejco.2021.100011>.
- [10] C. Audet : A survey on direct search methods for blackbox optimization and their applications. In P. Pardalos et T. Rassias, édés : *Mathematics without boundaries : Surveys in interdisciplinary research*, chap. 2, p. 31–56. Springer, New York, NY, 2014. URL <http://www.springer.com/mathematics/analysis/book/978-1-4939-1123-3>.
- [11] C. Audet, A. Brilli, Y. Diouane, S. Le Digabel, E. Silva et C. Tribes : A penalty-interior point method combined with MADS for equality and inequality constrained optimization. Rap. tech. 2601.20811, arXiv, 2026. URL <https://doi.org/10.48550/arXiv.2601.20811>.
- [12] C. Audet et J. Dennis, Jr. : Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006. URL <https://doi.org/10.1137/040603371>.

- [13] C. Audet et J. Dennis, Jr. : A Progressive Barrier for Derivative-Free Nonlinear Programming. *SIAM Journal on Optimization*, 20(1):445–472, 2009. URL <https://doi.org/10.1137/070692662>.
- [14] C. Audet, J. Dennis, Jr. et S. Le Digabel : Globalization strategies for Mesh Adaptive Direct Search. *Computational Optimization and Applications*, 46(2):193–215, 2010. URL <https://doi.org/10.1007/s10589-009-9266-1>.
- [15] C. Audet et W. Hare : *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2017. URL <https://doi.org/10.1007/978-3-319-68913-5>.
- [16] C. Audet, S. Le Digabel et M. Peyrega : Linear equalities in blackbox optimization. *Computational Optimization and Applications*, 61(1):1–23, 2015. URL <https://doi.org/10.1007/s10589-014-9708-2>.
- [17] C. Audet, S. Le Digabel, V. Rochon Montplaisir et C. Tribes : Algorithm 1027 : NOMAD version 4 : Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software*, 48(3):35 :1–35 :22, 2022. URL <https://doi.org/10.1145/3544489>.
- [18] C. Audet, S. Le Digabel et C. Tribes : The Mesh Adaptive Direct Search Algorithm for Granular and Discrete Variables. *SIAM Journal on Optimization*, 29(2):1164–1189, 2019. URL <https://doi.org/10.1137/18M1175872>.
- [19] S. Axen, M. Baran, R. Bergmann et K. Rzecki : Manifolds.jl : An Extensible Julia Framework for Data Analysis on Manifolds. *ACM Transactions on Mathematical Software*, 49(4), 2023. URL <https://doi.org/10.1145/3618296>.
- [20] N. Bartoli, T. Lefebvre, S. Dubreuil, R. Olivanti, R. Priem, N. Bons, J. Martins et J. Morlier : Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and Technology*, 90:85–102, 2019. URL <https://doi.org/10.1016/j.ast.2019.03.041>.
- [21] R. Bergmann : Manopt.jl : Optimization on Manifolds in Julia. *Journal of Open Source Software*, 7(70), 2022. URL <https://doi.org/10.21105/joss.03866>.
- [22] N. Boumal : *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023. URL <https://doi.org/10.1017/9781009166164>.
- [23] A. Boutot : Modernité de la catoptrique de Héron d’Alexandrie. *Philosophie antique*, 12:157–196, 2012. URL <https://doi.org/10.4000/philosant.938>.
- [24] A. Brilli, G. Liuzzi et S. Lucidi : An interior-point method for nonlinear constrained derivative-free optimization. *Optimization Methods and Software*, 40(3):611–649, 2025. URL <https://doi.org/10.1080/10556788.2025.2453110>.

- [25] L. Bueno, A. Friedlander, J. Martínez et F. Sobral : Inexact restoration method for derivative-free optimization with smooth constraints. *SIAM Journal on Optimization*, 23(2):1189–1213, 2013. URL <https://doi.org/10.1137/110856253>.
- [26] B. Cavaretta, F. Goyens, C. Royer et F. Yger : Complexity guarantees and polling strategies for Riemannian direct-search methods. Rap. tech. 2511.15360, arXiv, 2025. URL <https://doi.org/10.48550/arXiv.2511.15360>.
- [27] F. Clarke : *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990. URL <https://doi.org/10.1137/1.9781611971309>.
- [28] P. Conejo, E. Karas et L. Pedroso : A trust-region derivative-free algorithm for constrained optimization. *Optimization Methods and Software*, 30(6):1126–1145, 2015. URL <https://doi.org/10.1080/10556788.2015.1026968>.
- [29] A. Conn, N. Gould et P. Toint : *Trust-Region Methods*. MPS-SIAM Series on Optimization. SIAM, 2000. URL <http://www.numerical.rl.ac.uk/trbook/trbook.html>.
- [30] A. Conn, K. Scheinberg et P. Toint : A Derivative Free Optimization Algorithm in Practice. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998. <http://perso.fundp.ac.be/~phtoint/pubs/TR98-11.ps>.
- [31] A. Conn, K. Scheinberg et L. Vicente : Global convergence of general derivative-free trust-region algorithms to first and second order critical points. *SIAM Journal on Optimization*, 20(1):387–415, 2009. URL <https://doi.org/10.1137/060673424>.
- [32] A. Custódio, K. Scheinberg et L. Vicente : Methodologies and software for derivative-free optimization. In T. Terlaky, M. Anjos et S. Ahmed, eds : *Advances and Trends in Optimization with Engineering Applications*, MOS-SIAM Book Series on Optimization, chap. 37. SIAM, Philadelphia, 2017. URL <http://www.mat.uc.pt/~lnv/papers/dfo-survey.pdf>.
- [33] M. Diniz-Ehrhardt, J. Martinez et L. Pedroso : Derivative-free methods for nonlinear programming with general lower-level constraints. *Journal of Computational and Applied Mathematics*, 30(1), 2011. URL <https://www.scielo.br/j/cam/a/kK997pgKTV8VZXRSYYmrHMP/?lang=en>.
- [34] D. Dreisigmeyer : Direct Search Methods on Reductive Homogeneous Spaces. *Journal of Optimization Theory and Applications*, 176(3):585–604, 2018. URL <https://doi.org/10.1007/s10957-018-1225-5>.
- [35] D. Dreisigmeyer : Direct search methods over Riemannian manifolds. Rap. tech. LA-UR-06-7416, Los Alamos National Laboratory, Los Alamos, USA, 2006.

- [36] D. Dreisigmeyer : Equality constraints, Riemannian manifolds and direct search methods. Rap. tech. LA-UR-06-7406, Los Alamos National Laboratory, Los Alamos, USA, 2006. URL http://www.optimization-online.org/DB_HTML/2007/08/1743.html.
- [37] D. Dreisigmeyer : Direct search algorithms over Lipschitz manifolds. Rap. tech. LA-UR-07-1073, Los Alamos National Laboratory, Los Alamos, USA, 2007.
- [38] K. Dzahini, F. Rinaldi, C. Royer et D. Zeffiro : Direct-search methods in the year 2025 : Theoretical guarantees and algorithmic paradigms. *EURO Journal on Computational Optimization*, 13:100110, 2025. URL <https://doi.org/10.1016/j.ejco.2025.100110>.
- [39] H. Federer : Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959. URL <https://doi.org/10.1090/S0002-9947-1959-0110078-1>.
- [40] E. Fermi et N. Metropolis : Numerical solution of a minimum problem. Los Alamos Unclassified Report LA-1492, Los Alamos National Laboratory, Los Alamos, USA, 1952.
- [41] P. Ferreira, E. Karas, M. Sachine et F. Sobral : Global convergence of a derivative-free inexact restoration filter algorithm for nonlinear programming. *Optimization*, 66(2):271–292, 2017. URL <https://doi.org/10.1080/02331934.2016.1263629>.
- [42] D. Ge, J. Liu, T. Liu, J. Tan et Y. Ye : Algorithm 1053 : SOLNP+ : A Derivative-Free Solver for Constrained Nonlinear Optimization. *ACM Transactions on Mathematical Software*, 50(4), 2024. URL <https://doi.org/10.1145/3699956>.
- [43] W. Hock et K. Schittkowski : *Test Examples for Nonlinear Programming Codes*, vol. 187 de *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, Germany, 1981.
- [44] T. Kolda, R. Lewis et V. Torczon : Optimization by direct search : New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003. URL <https://doi.org/10.1137/S003614450242889>.
- [45] V. Kungurtsev, F. Rinaldi et D. Zeffiro : Retraction-Based Direct Search Methods for Derivative Free Riemannian Optimization. *Journal of Optimization Theory and Applications*, 203(2):1710–1735, 2024. URL <https://doi.org/10.1007/s10957-023-02268-3>.
- [46] J. Larson, M. Menickelly et S. Wild : Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019. URL <https://doi.org/10.1017/S0962492919000060>.
- [47] R. Lewis et V. Torczon : A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12(4):1075–1089, 2002. URL <https://doi.org/10.1137/S1052623498339727>.

- [48] R. Lewis et V. Torczon : A direct search approach to nonlinear programming problems using an augmented lagrangian method with explicit treatment of linear constraints. Rap. tech., College of William and Mary, 2010. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?Doi=10.1.1.156.8677>.
- [49] G. Liuzzi, S. Lucidi et M. Sciandrone : Sequential penalty derivative-free methods for nonlinear constrained optimization. *SIAM Journal on Optimization*, 20(5):2614–2635, 2010. URL <https://doi.org/10.1137/090750639>.
- [50] M. Lubin, O. Dowson, J. Dias Garcia, J. Huchette, B. Legat et J. Vielma : JuMP 1.0 : Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 15:581–589, 2023. URL <https://doi.org/10.1007/s12532-023-00239-3>.
- [51] S. Lucidi, M. Sciandrone et P. Tseng : Objective-derivative-free methods for constrained optimization. *Mathematical Programming*, 92(1):37–59, 2002. URL <https://doi.org/10.1007/s101070100266>.
- [52] J. Martínez et F. Sobral : Constrained derivative-free optimization on thin domains. *Journal of Global Optimization*, 56(3):1217–1232, 2013. URL <https://doi.org/10.1007/s10898-012-9944-x>.
- [53] J. Moré et S. Wild : Benchmarking Derivative-Free Optimization Algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009. URL <https://doi.org/10.1137/080724083>.
- [54] S. Na : Derivative-Free Sequential Quadratic Programming for Equality-Constrained Stochastic Optimization. Rap. tech. 2510.22458, arXiv, 2025. URL <https://doi.org/10.48550/arXiv.2510.22458>.
- [55] J. Nelder et R. Mead : A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, 1965. URL <http://comjnl.oxfordjournals.org/content/7/4/308.abstract>.
- [56] J. Nocedal et S. Wright : *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Berlin, second édn, 2006. URL <http://www.springer.com/mathematics/book/978-0-387-30303-1>.
- [57] S. Oddsen : The Riemannian LTMADS to solve the Spectral Procrustes problem in Manopt.jl. Mémoire de D.E.A., Norwegian University of Science and Technology, 2025. URL <https://nva.sikt.no/registration/0198ebfe15fd-b7944950-5a6a-4e57-b004-7173ec5d7344>. Available at <https://nva.sikt.no/registration/0198ebfe15fd-b7944950-5a6a-4e57-b004-7173ec5d7344>.
- [58] E. Omojokun : *Trust region algorithms for optimization with nonlinear equality and inequality constraints*. Thèse de doctorat, University of Colorado Boulder, 1989.

- [59] D. Platt et R. Sánchez Galán : Lower bound for the reach and applications. Rap. tech. 2505.08427, arXiv, 2025. URL <https://doi.org/10.48550/arXiv.2505.08427>.
- [60] M. Powell : A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In S. Gomez et J.-P. Hennart, édés : *Advances in Optimization and Numerical Analysis*, vol. 275 de *Mathematics and Its Applications*, p. 51–67. Springer, 1994. URL https://doi.org/10.1007/978-94-015-8330-5_4.
- [61] M. Powell : A View of Algorithms for Optimization Without Derivatives. Available at https://www.damtp.cam.ac.uk/user/na/NA_papers/NA2007_03.pdf, 2007.
- [62] T. Ragonneau : *Model-Based Derivative-Free Optimization Methods and Software*. Thèse de doctorat, The Hong Kong Polytechnic University, 2022. URL <https://theses.lib.polyu.edu.hk/handle/200/12294>. Available at <https://theses.lib.polyu.edu.hk/handle/200/12294>.
- [63] P. Sampaio et P. Toint : A derivative-free trust-funnel method for equality-constrained nonlinear optimization. *Computational Optimization and Applications*, 61(1):25–49, 2015. URL <https://doi.org/10.1007/s10589-014-9715-3>.
- [64] S. Selvan, P. Borckmans, A. Chattopadhyay et P.-A. Absil : Spherical mesh adaptive direct search for separating quasi-uncorrelated sources by range-based independent component analysis. *Neural Computation*, 25(9):2486–2522, 2013. URL https://doi.org/10.1162/NECO_a_00485.
- [65] T. Taminiau, E. Massart et G. Grapiglia : Riemannian optimization with finite-difference gradient approximations. Rap. tech. 2601.08751, arXiv, 2026. URL <https://doi.org/10.48550/arXiv.2601.08751>.
- [66] A. Tröltzsch : A sequential quadratic programming algorithm for equality-constrained optimization without derivatives. *Optimization Letters*, 10(2):383–399, 2016. URL <https://doi.org/10.1007/s11590-014-0830-y>.
- [67] A. Tröltzsch, Č. Illić et M. Siggel : SQPDFO – A trust-region based algorithm for generally-constrained derivative-free optimization. In *AIP Conference Proceedings*, vol. 2522. AIP Publishing, 2022. URL <https://doi.org/10.1063/5.0105349>.
- [68] M. Xi, W. Sun et J. Chen : Survey of derivative-free optimization. *Numerical Algebra, Control and Optimization*, 10(4):537–555, 2020. URL <https://doi.org/10.3934/naco.2020050>.
- [69] M. Xuan et J. Nocedal : A feasible method for constrained derivative-free optimization. *Operations Research Letters*, 65, 2025. URL <https://www.sciencedirect.com/science/article/pii/S0167637725001592>.

- [70] Z. Zhang : PRIMA : Reference Implementation for Powell's Methods with Modernization and Amelioration. available at <http://www.libprima.net>, DOI:10.5281/zenodo.8052654, 2023.