Les Cahiers du GERAD ISSN: 0711–2440

Optimisation non lisse avec contraintes: une approche par programmation quadratique séquentielle

B. Pineau, D. Orban

G–2026–69

October 2026

La collection *Les Cahiers du GERAD* est constituée des travaux de recherche menés par nos membres. La plupart de ces documents de travail a été soumis à des revues avec comité de révision. Lorsqu'un document est accepté et publié, le pdf original est retiré si c'est nécessaire et un lien vers l'article publié est ajouté.

Citation suggérée : B. Pineau, D. Orban (Octobre 2026). Optimisation non lisse avec contraintes: une approche par programmation quadratique séquentielle, Rapport technique, Les Cahiers du GERAD G– 2026–69, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2026-69) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

The series *Les Cahiers du GERAD* consists of working papers carried out by our members. Most of these pre-prints have been submitted to peer-reviewed journals. When accepted and published, if necessary, the original pdf is removed and a link to the published article is added.

Suggested citation: B. Pineau, D. Orban (October 2026). Optimisation non lisse avec contraintes: une approche par programmation quadratique séquentielle, Technical report, Les Cahiers du GERAD G–2026–69, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2026-69) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2026
– Bibliothèque et Archives Canada, 2026

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2026
– Library and Archives Canada, 2026

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Optimisation non lisse avec contraintes: une approche par programmation quadratique séquentielle

Benjamin Pineau [a]

Dominique Orban [a]

[a] *GERAD, Montréal (Qc), Canada, H3T 1J4*

bpineau59@gmail.com
dominique.orban@polymtl.ca

October 2026
Les Cahiers du GERAD
G–2026–69

Copyright © 2026 Pineau, Orban

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs. Les auteurs conservent leur droit d'auteur et leurs droits moraux sur leurs publications et les utilisateurs s'engagent à reconnaître et respecter les exigences légales associées à ces droits. Ainsi, les utilisateurs:

- Peuvent télécharger et imprimer une copie de toute publication du portail public aux fins d'étude ou de recherche privée;
- Ne peuvent pas distribuer le matériel ou l'utiliser pour une activité à but lucratif ou pour un gain commercial;
- Peuvent distribuer gratuitement l'URL identifiant la publication.

Si vous pensez que ce document enfreint le droit d'auteur, contactez-nous en fournissant des détails. Nous supprimerons immédiatement l'accès au travail et enquêterons sur votre demande.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*. Copyright and moral rights for the publications are retained by the authors and the users must commit themselves to recognize and abide the legal requirements associated with these rights. Thus, users:

- May download and print one copy of any publication from the public portal for the purpose of private study or research;
- May not further distribute the material or use it for any profit-making activity or commercial gain;
- May freely distribute the URL identifying the publication.

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Abstract : We consider the class of nonsmooth regularized optimization problems with equality constraints. Such problems arise in signal processing, machine learning, and low-rank matrix completion, where the nonsmooth term promotes structural properties such as sparsity or low total variation. We propose an extension of the Sequential Quadratic Programming (SQP) method introduced by Curtis, Qu and Robinson (2026) that handles a broad class of nonsmooth regularizers beyond the ℓ_1 norm. At each iteration, the tangent subproblem is solved using the Envelopt solver combined with a null-space reduction, which exploits the Moreau envelope to handle the nonsmooth term while enforcing the linearized constraints exactly. We further show that solving the tangent subproblem inexactly preserves the convergence and complexity results, provided that simple and sufficient decrease conditions are satisfied. Numerical experiments on problems from the CUTEst library confirm that the proposed method successfully handles various convex regularizers, and that inexact tangent step resolution significantly reduces the total number of inner solver iterations while maintaining solution quality.

Keywords : Nonsmooth optimization ; SQP methods ; equality-constrained optimization

Résumé : Nous considérons la classe de problèmes d'optimisation non lisse avec contraintes d'égalité. Ces problèmes apparaissent notamment en traitement du signal, en apprentissage automatique et pour la complétion de matrices de rang faible, où le terme non lisse permet de promouvoir des propriétés structurelles telles que la parcimonie ou la faible variation totale. Nous proposons une extension de la méthode de programmation quadratique séquentielle (SQP) introduite par Curtis, Qu et Robinson (2026) permettant de traiter une large classe de régulariseurs non lisses, au-delà de la seule norme ℓ_1. A chaque itération, le sous-problème tangent est résolu à l'aide du solveur Envelopt combiné à une méthode du noyau, qui exploite l'enveloppe de Moreau pour traiter le terme non lisse tout en satisfaisant exactement les contraintes linéarisées. Sous certaines conditions, nous montrons de plus que la résolution inexacte du sous-problème tangent préserve les résultats de convergence et de complexité. Des expériences numériques sur des problèmes issus de la bibliothèque CUTEst confirment que la méthode proposée résout efficacement des problèmes faisant intervenir divers régulariseurs convexes, et que la résolution inexacte du pas tangent réduit significativement le nombre total d'itérations du sous-solveur tout en préservant la qualité des solutions obtenues.

Mots clés : Optimisation non lisse ; méthodes SQP ; optimisation avec contraintes d'égalité

1 Introduction

Nous considérons la classe de problèmes d'optimisation non lisses avec contraintes

$$\min_{x\in\mathbb{R}^n} f(x)+h(x) \quad \text{tel que} \quad c(x)=0, \quad x\in\Omega, \tag{1}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ est continûment différentiable, $h : \mathbb{R}^n \longrightarrow \mathbb{R}\cup\{+\infty\}$ est propre, semi-continue inférieurement et convexe et $c : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ est continûment différentiable avec $m < n$. La résolution de problèmes de la forme (1) intervient dans de nombreuses applications en traitement du signal et problèmes inverses [11, 45], apprentissage automatique [46] ou encore pour la complétion de matrices de rang faible [10]. Dans ces contextes, le terme différentiable f modélise généralement la fidélité aux données tandis que le terme convexe non différentiable h est utilisé pour promouvoir certaines propriétés recherchées de la solution, telles que la parcimonie, la faible variation totale ou encore le faible rang. Ces modèles d'optimisation composite sont aujourd'hui omniprésents dans les applications modernes de l'optimisation numérique. Malgré leur large domaine d'application, ces problèmes demeurent difficiles à résoudre en raison de la combinaison de trois difficultés majeures : la présence de contraintes non linéaires, la non-différentiabilité du terme h et, dans de nombreuses applications, la grande dimension des problèmes considérés. Les méthodes classiques d'optimisation non linéaire supposent généralement que la fonction objectif est continûment différentiable et ne sont donc plus directement applicables.

Lorsque la fonction objectif comporte un terme non différentiable, plusieurs stratégies ont été proposées dans la littérature. Les méthodes de régularisation quadratique pour l'optimisation non lisse exploitent la structure composite du problème et permettent de conserver explicitement le terme non lisse grâce au calcul d'opérateurs proximaux [1, 19]. Elles sont particulièrement efficaces pour les problèmes non contraints. Il est possible d'inclure des contraintes dans la fonction non lisse h [37], notamment pour traiter les contraintes de bornes, mais leur extension aux contraintes non linéaires générales est nettement plus délicate. De plus la vitesse de convergence des multiplicateurs est linéaire en général [3, Section 2.2], ce qui peut ralentir la convergence globale pour résoudre le problème initial.

Une autre approche repose sur les méthodes de Lagrangien augmenté, qui remplacent le problème original par une suite de sous-problèmes dans lesquels les contraintes sont ajoutées à la fonction objectif, tout en mettant à jour des multiplicateurs de Lagrange [24]. Il s'agit alors de résoudre une série de problèmes non contraints. Ces méthodes bénéficient d'un cadre théorique solide et constituent aujourd'hui l'une des approches les plus générales pour la résolution de problèmes contraints. Néanmoins, leur efficacité dépend de la résolution des sous-problèmes internes, qui peuvent devenir coûteux lorsque les contraintes sont nombreuses ou mal conditionnées [33, Section 17.3]. Initialement, les méthodes de Lagrangien augmenté ont été présentées pour résoudre des problèmes lisses, mais des modifications ont été proposées pour traiter des problèmes composites non-différentiables avec contraintes [16].

Dans le cas où $h \equiv 0$, les méthodes de programmation quadratique séquentielle (Sequential Quadratic Programming en anglais, que nous noterons par la suite SQP) constituent aujourd'hui l'une des approches les plus performantes pour la résolution de problèmes d'optimisation non linéaires sous contraintes [7, 25, 33]. A chaque itération, elles construisent une approximation quadratique locale de la fonction objectif ainsi qu'une approximation linéaire des contraintes, conduisant à la résolution d'une succession de sous-problèmes quadratiques. Ces méthodes possèdent d'excellentes propriétés de convergence locale [33, Section 18.2] et constituent le fondement de solveurs de référence tels que KNITRO [9]. Contrairement aux méthodes de Lagrangien, les méthodes SQP démontrent leur efficacité lors de la résolution de problèmes présentant des contraintes fortement non linéaires [33, Section 18.1].

Afin de combiner les avantages des méthodes SQP et des approches de régularisation quadratique, Curtis et al. [14] proposent une méthode SQP adaptée aux problèmes composites. A l'instar de certaines méthodes SQP, chaque itération est décomposée en un pas normal visant à réduire la violation des contraintes, puis un pas tangent destiné à améliorer la fonction objectif. Leur approche conserve cependant explicitement le terme convexe non différentiable dans le sous-problème tangent tout en utilisant un modèle de la partie lisse. En revanche, numériquement, le terme non lisse est

remplacé par un terme lisse en introduisant des contraintes supplémentaires dans le problème initial. Aucun résultat numérique n'est présenté pour les problèmes dont le terme non lisse ne peut pas être reformulé à l'aide d'un terme lisse et l'ajout de contraintes.

Dans ce contexte, l'objectif de ce stage est de développer un solveur efficace pour la résolution du problème (1) pour divers régulariseurs convexes autres que la norme ℓ_1. Pour cela, nous nous appuyons sur le cadre SQP proposé par Curtis et al. [14], tout en résolvant le sous-problème tangent à l'aide du solveur ENVELOPT [18]. Ce dernier repose sur une méthode de Lagrangien augmenté permettant de résoudre des problèmes d'optimisation composites. Afin de traiter efficacement les contraintes linéarisées, le problème tangent est reformulé à l'aide d'une méthode du noyau [25, 33], permettant de travailler directement dans l'espace tangent aux contraintes. Enfin, contrairement aux travaux de Curtis et al. [14], qui se concentrent uniquement sur la norme ℓ_1, nous étendons cette approche à une famille plus générale de fonctions convexes non différentiables et évaluons expérimentalement ses performances sur différentes classes de problèmes.

Dans la section 2, nous aborderons les fondements des méthodes de Lagrangien augmenté, de la programmation quadratique séquentielle ainsi que des méthodes de régularisation quadratique non lisse. Nous présenterons également l'algorithme ENVELOPT. Dans la section 3, nous présenterons dans un premier temps un algorithme permettant de résoudre un problème quadratique non lisse avec contraintes. Par la suite, nous proposerons une adaptation de l'algorithme introduit dans [14] afin de résoudre le problème (1). Enfin, nous discuterons de la nécessité de la résolution exacte du problème quadratique étudié dans la section 3.1. La section 4 illustrera les résultats numériques des algorithmes présentés dans la section 3. Nous terminerons ce rapport par une conclusion générale ainsi que par une ouverture sur les perspectives envisagées à la suite de ces travaux.

2 Contexte et état de l'art

2.1 Algorithmes itératifs

Afin de résoudre des problèmes d'optimisation, il est courant de se reposer sur une approche itérative consistant à remplacer, à chaque itération, le problème initial par un sous-problème plus simple à résoudre [33]. En notant x_k l'itéré courant, le sous-problème est alors construit à partir des informations disponibles en x_k, telles que la valeur de la fonction objectif, son gradient ou encore une approximation de sa Hessienne [7, 12, 33]. La résolution de ce sous-problème permet de déterminer une nouvelle direction ou un nouvel itéré x_{k+1}. Ce dernier sert ensuite de point de référence pour construire le sous-problème de l'itération suivante. On obtient ainsi une suite d'itérés $(x_k)_{k\geq 0}$ dont l'objectif est de converger vers une solution du problème initial. La qualité du sous-problème considéré ainsi que le degré de précision avec lequel il est résolu jouent alors un rôle central dans l'efficacité de la méthode. Les méthodes de Lagrangien, SQP ou de régularisation quadratique non lisse que nous aborderons dans ce rapport s'appuient toutes sur ce principe.

2.2 Problème avec contraintes et Lagrangien

Nous nous intéressons dans cette partie à la résolution de problème lisse avec contraintes d'égalité (2)

$$\min_x f(x) \quad \text{tel que} \quad c(x) = 0 \tag{2}$$

en reformulant le problème et en incluant les contraintes dans la fonction objectif.

2.2.1 Fonction Lagrangien et stationnarité

En introduisant le Lagrangien du problème défini par

$$\mathcal{L}(x, y) = f(x) - y^\top c(x), \tag{3}$$

alors sous certaines hypothèses de régularité, la résolution du problème (2) peut être caractérisée, au premier ordre, par les conditions de stationnarité du Lagrangien [33, Chap. 12]. La quantité y est appelée multiplicateur de Lagrange. En notant $x_\star$ une solution locale de (2), en supposant que $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ et $c : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ sont des fonctions lisses, et que la jacobienne des contraintes en $x^\star$ soit de rang plein, alors il existe des multiplicateurs de Lagrange $y^\star$ [33] tels que :

$$\begin{cases} \nabla_x \mathcal{L}(x_\star, y_\star) = 0 & \text{(4a)} \\ c(x_\star) = 0 & \text{(4b)} \end{cases}$$

Ce système d'équations constitue une *condition nécessaire du premier ordre*, également appelée *conditions de Karush–Kuhn–Tucker (KKT)*. L'équation (4a) implique que le couple $(x_\star, y_\star)$ est bien un point stationnaire de (2). L'équation (4b) permet de vérifier que le point $x_\star$ est un point réalisable. Le système d'équations (4) nous sera utile par la suite car il nous permettra de savoir si le point auquel nous nous trouvons à l'itération k est un minimum local.

2.2.2 Lagrangien augmenté

La méthode du Lagrangien permet de prendre en compte les contraintes en les incorporant directement dans la fonction objectif à l'aide de multiplicateurs de Lagrange. Cependant, en pratique, la résolution du système des conditions KKT nécessite de satisfaire simultanément la stationnarité (4a) et la faisabilité (4b). Une autre approche consiste à pénaliser la violation des contraintes, par exemple au moyen d'une pénalisation quadratique [13, 23]. Cette dernière permet de traiter la contrainte en transformant le problème en problème sans contraintes, mais nécessite généralement d'augmenter progressivement un paramètre de pénalisation μ, ce qui peut conduire à des problèmes de mauvais conditionnement lorsque μ devient grand. Le Lagrangien augmenté combine ces deux approches en associant un terme de Lagrangien à une pénalisation quadratique des contraintes. Il permet ainsi de contrôler simultanément la violation des contraintes et l'estimation des multiplicateurs de Lagrange, sans nécessiter des valeurs excessivement grandes du paramètre de pénalisation [3, 33]. Le Lagrangien augmenté est défini par

$$\mathcal{L}(x, y; \mu) = f(x) - \sum_{i=1}^{m} y_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^{m} c_i^2(x). \tag{5}$$

Nous remarquons l'influence de pénalité quadratique par la présence du terme $\frac{\mu}{2} \sum_{i=1}^{m} c_i^2(x)$, mais également du Lagrangien de par la présence du terme $\sum_{i=1}^{m} y_i c_i(x)$. La condition d'optimalité devient alors (voir [33, Equation (17.37)])

$$0 \approx \nabla_x \mathcal{L}(x_k, y^k; \mu_k) = \nabla f(x) - \sum_{i=1}^{m} \left[y_i^k - \mu_k c_i(x_k) \right] \nabla c_i(x_k).$$

En comparaison avec les conditions du premier ordre KKT classique (4), nous avons

$$\begin{aligned} & \forall i,\ y_i^\star \approx y_i^k - \mu_k c_i(x_k) \\ \iff & \forall i,\ c_i(x_k) \approx \frac{-1}{\mu_k}(y_i^\star - y_i^k). \end{aligned} \tag{6}$$

L'algorithme général permettant de résoudre (2) est présenté en Annexe C.1, Algorithme C1. Le Lagrangien augmenté offre deux manières d'augmenter la précision de l'estimé $x_\star$, solution du problème (2) : il est possible d'augmenter la valeur de μ_k, mais disposer d'une estimation suffisamment précise des multiplicateurs de Lagrange y^k permet également d'augmenter la précision [33, Théorème 17.6]. Dans la formulation avec pénalisation quadratique, nous ne pouvons qu'augmenter μ_k. Dans la formulation avec uniquement le Lagrangien, nous devons obtenir des multiplicateurs de manière très précise. La méthode de Lagrangien augmenté tire parti des avantages du Lagrangien et de la pénalisation quadratique, mais ne permet de résoudre que des problèmes lisses avec contraintes et ne traite pas directement le terme non lisse h présent dans le problème (1).

2.3 Programmation quadratique séquentielle

L'une des méthodes les plus efficaces présentes dans la littérature pour résoudre des problèmes non linéaire avec contraintes de la forme

$$\min_x f(x) \quad \text{tel que} \quad c(x) = 0 \tag{7}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ et $c : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ sont des fonctions lisses consiste à générer des pas en résolvant des sous-problèmes quadratiques. Nous décrirons dans cette section les principes de cette méthode appelée programmation quadratique séquentielle [5, 44] (Sequential Quadratic Programming en anglais, notée par la suite SQP).

2.3.1 Principe général

L'idée principale de la méthode SQP pour résoudre le problème d'optimisation est de modéliser (7) à l'itéré x_k par un modèle quadratique, puis d'utiliser le minimiseur de ce sous-problème pour définir l'itéré suivant x_{k+1}. Le Lagrangien (voir (3)) du problème (7) s'écrit $\mathcal{L}(x, y) = f(x) - y^\top c(x)$. Les conditions KKT (voir (4)) du premier ordre donnent

$$F(x, y) = \begin{bmatrix} \nabla f(x) - J(x)^\top y \\ c(x) \end{bmatrix} = 0. \tag{8}$$

Une manière de résoudre ce système d'équations est d'utiliser la méthode de Newton [5, 33]. Le pas de Newton à l'itéré (x_k, y_k) est défini par

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_y \end{bmatrix}$$

où les directions de Newton (p_k, p_y) vérifient

$$\begin{aligned} F'(x_k, y_k) \begin{bmatrix} p_k \\ p_y \end{bmatrix} &= -F(x_k, y_k) \\ \iff \begin{bmatrix} \nabla^2_{x,x}\mathcal{L}_k & -J_k^\top \\ J_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ p_y \end{bmatrix} &= \begin{bmatrix} -\nabla f_k + J_k^\top y_k \\ -c_k \end{bmatrix}. \end{aligned} \tag{9}$$

En introduisant les nouveaux itérés

$$x_{k+1} = x_k + p_k, \qquad l_k = y_k + p_y,$$

la première équation du système de Newton devient

$$\nabla^2_{x,x}\mathcal{L}_k p_k - J_k^\top p_y = -\nabla f_k + J_k^\top y_k.$$

En utilisant la relation $p_y = l_k - y_k$, nous obtenons

$$\nabla^2_{x,x}\mathcal{L}_k p_k - J_k^\top (l_k - y_k) = -\nabla f_k + J_k^\top y_k,$$

et finalement

$$\nabla^2_{x,x}\mathcal{L}_k p_k + \nabla f_k - J_k^\top l_k = 0.$$

De même, la seconde équation du système de Newton donne

$$J_k p_k + c_k = 0.$$

Ainsi, les conditions de Newton peuvent s'écrire

$$\begin{bmatrix} \nabla^2_{x,x}\mathcal{L}_k p_k + \nabla f_k - J_k^\top l_k \\ J_k p_k + c_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

qui n'est autre que l'expression des conditions KKT du premier ordre pour le problème

$$\min_p \; f_k + \nabla f_k^\top p + \frac{1}{2} p^\top \nabla^2_{x,x} \mathcal{L}_k p \quad \text{tel que} \quad J_k p + c_k = 0. \tag{10}$$

Ainsi les itérés générés par la méthode de Newton sont solutions du sous-problème quadratique (10) et l'Algorithme 1 permet de résoudre le problème (7) :

Algorithme 1 Algorithme SQP local pour résoudre (10)

Entrée : Couple initial (p_0, y_0)
1: Initialiser $k = 0$
2: **Tant que** Test de convergence non satisfait **faire :**
3: Calculer f_k, ∇f_k, $\nabla^2_{x,x}\mathcal{L}_k$ et J_k
4: Résoudre le problème (10) pour obtenir (p_k, l_k)
5: Mettre à jour $x_{k+1} = x_k + p_k$, $y_{k+1} = l_k$
6: **Fin**
7: **Sortie :** (x_k, y_k)

Remarque 1. la quantité f_k étant indépendante de p, chercher le minimiseur de $f_k + \nabla f_k^\top p + \frac{1}{2} p^\top \nabla^2_{x,x} \mathcal{L}_k p$ est équivalent à chercher le minimiseur de $\nabla f_k^\top p + \frac{1}{2} p^\top \nabla^2_{x,x} \mathcal{L}_k p$.

L'approche induite par la formulation (9) est la résolution du système par méthode directe. Cette approche consiste à résoudre directement le système linéaire de type KKT du premier ordre afin d'obtenir simultanément les directions p_k et p_y [5, 8]. Résoudre ce type de système linéaire peut être coûteux en temps de calcul et en mémoire lorsque le nombre de variables ou de contraintes est important. Au contraire, l'approche induite par la formulation (10) est la résolution d'un problème d'optimisation quadratique contraint. Cette formulation permet de se concentrer directement sur la recherche de la direction p_k notamment via une résolution itérative [14, 15, 25], souffrant moins des désavantages présentés précédemment. Nous présentons une méthode s'appuyant sur la formulation (10) dans la partie suivante. Notons que les deux formulations sont tout de même étroitement liées : les conditions d'optimalité du premier ordre du problème quadratique (10) conduisent précisément au système (9).

2.3.2 Découplage des pas tangent et normal

Il existe une variété de méthodes SQP (voir Nocedal et Wright [33, Chap. 18]) pour résoudre (7). La résolution du sous-problème quadratique doit à la fois permettre de réduire la violation des contraintes et d'améliorer la fonction objectif. Afin de traiter séparément ces deux aspects, nous nous intéressons aux méthodes qui décomposent chaque sous-problème SQP en deux sous-problèmes [7, 25, 34]. Le premier sous-problème, représenté par (11), est appelé sous-problème normal (ou vertical) et permet d'améliorer la faisabilité des contraintes :

$$\begin{aligned} \min_v \quad & \|J_k v + c_k\|_2^2 \\ \text{tel que} \quad & \|v\|_2 \leq \xi \Delta_k, \quad \xi \in (0,1) \end{aligned} \tag{11}$$

Nous noterons v_k la solution du problème (11). L'objectif est alors de calculer un pas s qui ne dégrade pas la réduction de violation des contraintes obtenue grâce à v_k tout en améliorant la fonction objectif :

$$\begin{aligned} \min_s \quad & \nabla f_k^\top s + \frac{1}{2} s^\top \nabla^2_{xx} \mathcal{L}_k s \\ \text{tel que} \quad & J_k s = J_k v_k \\ & \|s\|_2 \leq \Delta_k \end{aligned} \tag{12}$$

Remarquons que le problème (12) possède toujours une solution réalisable (en effet, $s = v_k$ est une solution satisfaisant les contraintes). Le pas s est calculé en trouvant un pas complémentaire à v_k

à partir d'un autre sous-problème. Ce deuxième sous-problème, représenté par le problème (13), est appelé problème tangent (ou horizontal) :

$$\begin{aligned} \min_{u\in\mathbb{R}^n} \quad & \left(\nabla f_k + \nabla^2_x \mathcal{L}_k v_k\right)^\top u + \frac{1}{2} u \nabla^2_x \mathcal{L}_k u \\ \text{tel que} \quad & J_k u = 0, \quad \|u\| \leq \sqrt{\Delta_k^2 - \|v_k\|^2}. \end{aligned} \tag{13}$$

En calculant les pas normal et tangent de cette manière, v_k et u_k sont alors orthogonaux. Nous pouvons accepter ou rejeter le pas $s_k = v_k + u_k$ grâce à différents critères, la plupart faisant intervenir une fonction de mérite. En cas d'acceptation, nous définissons le nouvel itéré par $x_{k+1} = x_k + s_k$ et nous utilisons ce point afin de modéliser (7) par un nouveau modèle quadratique.

Cependant cette approche, telle que présentée dans [7, 25, 33], ne concerne que les problèmes lisses (la fonction objectif et les contraintes sont supposées au minimum appartenir à $\mathcal{C}^1(\mathbb{R}^n)$). Nous souhaitons cependant résoudre des problèmes ayant un terme non lisse (voir problème (1)). Curtis et al. [14] proposent une modification de la méthode SQP avec découplage des pas tangent et normal pour traiter les problèmes (1). Néanmoins, les tests numériques de l'article ne portent que sur $h(x) = \lambda\|x\|_1$, où la norme est transformée en fonction lisse (voir Annexe B), rendant le problème original d'intérêt lisse. Or, notre objectif est de pouvoir traiter une classe beaucoup plus large de fonctions non lisses, plutôt que de nous limiter au cas particulier où la norme ℓ_1 est régularisée.

2.4 Méthodes de régularisation quadratique

Nous considérons la classe de problèmes non lisses

$$\min_x \; f(x) + h(x) \tag{14}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ est continûment différentiable et $h : \mathbb{R}^n \longrightarrow \mathbb{R} \cup \{+\infty\}$ est propre et semi-continue inférieurement (voir [39, Définition 1.5]). Des stratégies de descente de gradient permettent de résoudre (14) comme la descente de gradient proximal [2] qui consiste à générer une suite d'itérés $(x_k)_{k\geq 0}$ telle que

$$x_{k+1} \in \operatorname{prox}_{\nu h}(x_k - \nu \nabla f(x_k)),$$

où ν est un paramètre proximal dont le rôle est similaire à celui du pas de descente dans une méthode de gradient classique. On pourra se référer à [39, Chap.1] pour la définition de l'opérateur proximal d'une fonction et les propriétés qui en découlent. Cependant des méthodes plus sophistiquées ont récemment émergé. Aravkin et al. [1] créent respectivement φ et ψ des modèles de f et h, où φ est un modèle linéaire de f tel que

$$\varphi(s; x) = f(x) + \nabla f(x)^\top s \approx f(x + s)$$

et $\psi(s; x) \approx h(x + s)$. A chaque itération, il s'agit alors de résoudre le problème régularisé

$$\min_s \; m(s; x_k, \sigma_k) = \varphi(s; x_k) + \psi(s; x_k) + \frac{1}{2}\sigma_k \|s\|_2^2 \tag{15}$$

où $\sigma_k = 1/\nu_k$ est un paramètre de régularisation. Le paramètre de régularisation est adapté selon le comportement local du problème, puis le nouvel itéré est défini par $x_{k+1} = x_k + s_k$ où s_k est la solution de (15). Nous pouvons faire le lien entre la méthode régularisée [1] et la descente de gradient proximal en remarquant que

$$\begin{aligned} & x_{k+1} \in \operatorname{prox}_{\nu_k h}\left(x_k - \nu_k \nabla f(x_k)\right) \\ \iff & x_{k+1} \in x_k + \operatorname*{argmin}_s \nabla f(x_k)^\top s + \frac{1}{2}\nu_k^{-1}\|s\|^2 + h(x_k + s) \\ \iff & x_{k+1} \in x_k + \operatorname*{argmin}_s f(x_k) + \nabla f(x_k)^\top s + \frac{1}{2}\sigma_k\|s\|^2 + h(x_k + s) \\ \iff & x_{k+1} = x_k + s_k, \quad s_k \in \arg\min_s \; m(s; x_k, \sigma_k). \end{aligned}$$

L'algorithme, appelé R2, est présenté en Annexe C.2, Algorithme C2. Aravkin et al. [1] montrent que cet algorithme obtient de meilleurs résultats qu'une descente de gradient classique. Diouane et al. [19] proposent d'utiliser le même mécanisme en construisant un modèle quadratique de f :

$$\varphi(x;s) = f(x) + \nabla f(x)^\top s + \frac{1}{2} s^\top B(x) s \approx f(x+s)$$

où $B(x) = B(x)^\top \in \mathbb{R}^{n\times n}$ peut être la vraie hessienne de f ou une approximation. L'algorithme, appelé R2N, est présenté en Annexe C.2, Algorithme C3. Notons que [19] utilise R2 [1] pour résoudre (15) lorsque le modèle est quadratique. Les méthodes de régularisation quadratique dans le cadre non lisse s'avèrent efficaces pour résoudre (14), cependant ces problèmes sont des problèmes sans contraintes, contrairement à (1). Il est possible d'inclure des contraintes de bornes dans la fonction non lisse h par l'ajout de fonctions indicatrice par exemple [37], mais leur extension aux contraintes non linéaires générales est nettement plus délicate.

2.5 Optimisation convexe composite avec contraintes

Nous introduisons dans cette partie la méthode Envelopt [18]. Cette méthode permet de résoudre des problèmes non lisses avec contraintes de la forme

$$\min_x f(x) + h(F(x)) \quad \text{tel que} \quad l \le c(x) \le u, \quad x \in \Omega \tag{16}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ et $F : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ sont continûment différentiables, $h : \mathbb{R}^p \longrightarrow \mathbb{R} \cup \{+\infty\}$ est propre, semi-continue inférieurement et convexe et $c : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ est continûment différentiable avec $m < n$. Pour ce faire, De Marchi et Orban [18] résolvent une suite de problèmes lisses avec contraintes, obtenue grâce aux transformations suivantes :

1. Soit $s = F(x)$, alors le problème à résoudre devient

$$\min_{x,s} f(x) + h(s), \quad s = F(x), \quad l \le c(x) \le u, \quad x \in \Omega$$

2. Le Lagrangien augmenté associé à ce nouveau problème est

$$\mathcal{L}(x,s;y,\mu) = f(x) + h(s) + \frac{1}{2\mu}\|F(x) - s + \mu y\|^2$$

   en ajoutant la constante (par rapport à x et s) $\frac{\mu}{2}\|y\|^2$
3. Il s'agit alors de résoudre la suite de problème

$$\min_{x,s} \mathcal{L}(x,s;y,\mu) \quad \text{tel que} \quad \ell \le c(x) \le u.$$

4. Dans un premier temps, De Marchi et Orban [18] minimisent selon s :

$$\begin{aligned}\min_s \mathcal{L}(x,s;y,\mu) &= f(x) + \min_s h(s) + \frac{1}{2\mu}\|F(x) - s + \mu y\|^2 \\ &= f(x) + h_\mu(F(x) + \mu y)\end{aligned}$$

   où h_μ est l'enveloppe de Moreau de h (voir [39, Définition 1.22]).
5. Il suffit alors de minimiser le problème selon x, et ainsi résoudre le problème (16) est équivalent à résoudre le problème (17)

$$\min_x f(x) + h_\mu\left(F(x) + \mu y\right) \quad \text{tel que} \quad l \le c(x) \le u, \quad x \in \Omega \tag{17}$$

Lorsque h est convexe, l'enveloppe de Moreau est continûment différentiable [39, Théorème 2.26]. Le problème (17) est donc un problème lisse avec contraintes pouvant être résolu par de nombreux solveurs

de la littérature comme IPOPT [43] ou TRON [28]. Bien qu'ENVELOPT permette de résoudre directement le problème (1), cette approche présente certaines limitations qui motivent le développement de la méthode proposée dans ce rapport. En effet, ENVELOPT s'appuie sur un Lagrangien augmenté, dont la convergence vers un point KKT dépend de la qualité de l'estimation des multiplicateurs de Lagrange. Les méthodes SQP en revanche, possèdent de bonnes propriétés de convergence locale. L'approche proposée vise donc à combiner les avantages des deux méthodes : la capacité d'ENVELOPT à traiter des termes non lisses grâce à l'enveloppe de Moreau, et l'efficacité de la méthode SQP.

3 Développement méthodologique

Notre objectif est de résoudre le problème d'optimisation non lisse (1). Pour ce faire, nous allons adapter la méthode SQP présentée dans [14]. Dans un premier temps, nous présentons une manière de résoudre le problème tangent en utilisant ENVELOPT. Par la suite, nous détaillerons l'algorithme permettant de résoudre le problème (1).

3.1 Résolution de problèmes quadratiques non lisses

Cette partie se consacre à la résolution de problèmes quadratiques de forme générale

$$\min_{s\in\mathbb{R}^n} \; g^\top s + \frac{1}{2} s^\top H s + \psi(s) \quad \text{tel que} \quad Js = 0, \tag{18}$$

où $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ est une fonction convexe non lisse représentant un modèle de la fonction h, $g \in \mathbb{R}^n$, $H \in \mathbb{R}^{n\times n}$ et $J \in \mathbb{R}^{m\times n}$. Le problème (18) est formulé de cette manière dans le but de résoudre des problèmes plus généraux à l'aide de la méthode SQP.

Nous souhaitons appliquer ENVELOPT au problème quadratique (18). Pour nous assurer d'avoir un point réalisable à chaque itération (c'est-à-dire que les contraintes soient respectées à chaque itération), nous nous plaçons dans l'espace tangent aux contraintes en x_k [25, 33]. En particulier, en notant $Z \in \mathbb{R}^{n\times(n-m)}$ une matrice dont les colonnes forment une base de $\ker(J)$, alors $JZ = 0$. Par conséquent, tout élément $s \in \mathbb{R}^n$ s'écrivant $s = Zt$ appartient à $\ker(J)$ et vérifie donc $Js = 0$. Tout vecteur de la forme Zt constitue ainsi un point réalisable du problème (18). Nous procédons au changement de variable $s = Zt$ et le problème (18) peut être réécrit en :

$$\min_t \varphi(Zt) + \psi(Zt), \tag{19}$$

où $\varphi(s) = g^\top s + \frac{1}{2} s^\top H s$. Avec la méthode ENVELOPT, en utilisant (17), le problème (19) est reformulé en :

$$\min_t \varphi(Zt) + \psi_\mu(Zt + \mu y). \tag{20}$$

Par hypothèse, ψ est convexe, donc l'enveloppe de Moreau est lisse et son gradient est (voir [39, Théorème 2.26]) :

$$\nabla\psi_\mu(x) = \mu^{-1}\left(x - \operatorname*{prox}_{\mu\psi}(x)\right).$$

Il découle que le problème (20) est lisse et peut être résolu par divers solveurs. Entre autre, nous pouvons citer TRON [1], TRUNK [2], IPOPT [35] [3] et MADNLP (voir respectivement [28], [12], [43] et [40, 41]) [4]. Finalement, l'Algorithme 2 permet de résoudre le problème quadratique (18). On note $\tilde{\varphi}(t) = \varphi(Zt)$ la fonction φ réduite à l'espace tangent.

1. JuliaSmoothOptimizers/JSOSolvers.jl/src/tron.jl
2. JuliaSmoothOptimizers/JSOSolvers.jl/src/trunk.jl
3. JuliaSmoothOptimizers/NLPModelsIpopt.jl
4. MadNLP/MadNLP.jl

Algorithme 2 Méthode Envelopt - Noyau pour résoudre (18).

1: Calcul de $Z =$ base de $\ker(J)$
2: Réduire à un sous-espace sans contrainte :

$$\tilde{\varphi}(t) \;=\; \varphi(Zt), \qquad F(t) \;=\; Zt$$

3: Résoudre le problème (20) avec ENVELOPT avec un sous-solveur au choix (IPOPT, TRUNK, TRON, MADNLP)
4: Retour à l'espace d'origine :

$$x_\star \;=\; Zt_\star$$

5: **Sortie :** $x_\star$

3.2 Méthode SQP pour l'optimisation non lisse sous contraintes

3.2.1 Algorithme

Afin de résoudre la classe de problèmes (1) où h est un régulariseur convexe quelconque, nous adaptons la méthode SQP présentée par Curtis et al. [14] en utilisant ENVELOPT comme présenté dans la section 3.1 pour résoudre le problème tangent (13). Détaillons les étapes de l'algorithme. Pour résoudre le problème (1), Curtis et al. [14] résolvent dans un premier temps un problème normal permettant d'améliorer la faisabilité. Le pas normal est défini en résolvant le problème

$$\begin{aligned} \min_{v} \quad & m_k(v) = \frac{1}{2}\|J_k v + c_k\|_2^2 \\ \text{tel que} \quad & \|v\|_2 \leq \kappa_v \nu_k \delta_k, \quad x_k + v \in \Omega \end{aligned} \tag{21}$$

où $\delta_k = \|\nabla_\Omega \varphi_c(x_k)\|_2 = \|\text{Proj}_{T_\Omega(x_k)}(-J_k^\top c_k)\|$, $\text{Proj}_{T_\Omega(x_k)}(-J_k^\top c_k)$ est la projection sur le cône tangent de Ω en x_k du vecteur $-J_k^\top c_k$ et $\varphi_c(x) = \frac{1}{2}\|c(x)\|_2^2$. Le problème (21) est une généralisation du problème (11) [7] [33] pour lequel nous avons en plus des contraintes de bornes $x \in \Omega$. Notons que le pas normal doit également satisfaire une décroissance suffisante du modèle déterminée par un pas de Cauchy v_k^c, de sorte que $m_k(v_k) \leq m_k(v_k^c)$. Nous ne détaillerons pas le calcul du pas de Cauchy ici, mais nous faisons remarquer que le pas de Cauchy est un pas facile à calculer qui induit une décroissance suffisante du modèle, utile dans la démonstration de convergence de l'algorithme. Après avoir résolu le problème normal (21), dont nous notons v_k la solution, Curtis et al. [14] résolvent un pas tangent, permettant d'améliorer la fonction objectif sans détériorer la faisabilité atteinte par v_k. Le pas tangent est défini en résolvant le problème

$$\begin{aligned} \min_{u} \quad & g_k^\top u + \frac{1}{2\nu_k}\|u\|_2^2 + \frac{1}{\nu_k} v_k^\top u + h(x_k + v_k + u) \\ \text{tel que} \quad & J_k u = 0, \quad x_k + v_k + u \in \Omega. \end{aligned} \tag{22}$$

Nous pouvons remarquer que, contrairement au problème tangent (13), le problème proposé par Curtis et al. [14] ne fait pas intervenir de région de confiance mais repose plutôt sur une régularisation quadratique, comme décrit dans la section 2.4. Le problème (22) n'utilise également pas l'information du second ordre.

Nous notons u_k la solution du problème tangent (22), et créons ainsi le pas candidat $s_k = v_k + u_k$. Ce pas candidat est retenu s'il induit une réduction suffisante de la fonction de mérite (23)

$$\phi_\tau(x) = \tau\,(f(x) + h(x)) + \|c(x)\|_2 \tag{23}$$

où τ est un paramètre ajusté au cours des itérations de l'algorithme de sorte que le pas candidat soit une direction de descente. Le pas s_k est accepté si la condition (24) est vérifiée :

$$\phi_{\tau_k}(x_k + s_k) - \phi_{\tau_k}(x_k) \leq -\eta_\phi \left(\frac{\tau_k}{4\nu_k}\|s_k\|_2^2 + \sigma_c(\|c_k\|_2 - \|c_k + J_k v_k\|_2) \right). \tag{24}$$

Algorithme 3 Méthode SQP pour résoudre (1).

Entrées : $x_0 \in \Omega$, $\{\nu_0, \tau_{-1}, \kappa_\tau, \kappa_v\} \subset \mathbb{R}_{>0}$, et $\{\xi, \eta_\Phi, \sigma_c, \varepsilon_\tau, \gamma, \eta_m\} \subset (0,1)$.
pour $k = 0, 1, 2, \ldots$ **faire :**
  Calculer δ_k.
  si $\delta_k = 0$ **alors**
    Poser $v_k \leftarrow 0$.
    si $\|c_k\|_2 \neq 0$ **alors**
      Sortie : x_k (point stationnaire non réalisable).
    fin si
  sinon
    Calculer v_k, solution approchée du problème (21) satisfaisant $m_k(v_k) \leq m_k(v_k^c)$.
  fin si
  Calculer u_k comme l'unique solution du sous-problème (22).
  Poser $s_k \leftarrow v_k + u_k$.
  si $\|s_k\|_2/\nu_k = 0$ **alors**
    Sortie : x_k (point KKT du premier ordre du problème (1)).
  fin si
  Calculer τ_k à l'aide de (E3).
  si $\phi_{\tau_k}(x_k + s_k) - \phi_{\tau_k}(x_k) \leq -\eta_\Phi \left(\frac{\tau_k}{4\nu_k} \|s_k\|_2^2 + \sigma_c \big(\|c_k\|_2 - \|c_k + J_k s_k\|_2 \big) \right)$ **alors**
    Poser $x_{k+1} \leftarrow x_k + s_k$ et $\nu_{k+1} \leftarrow \nu_k$.
  sinon
    Poser $x_{k+1} \leftarrow x_k$ et $\nu_{k+1} \leftarrow \xi \nu_k$.
  fin si
fin pour

La mise à jour du paramètre τ est motivée et détaillée en Annexe E. L'Algorithme 3 permet alors de résoudre le problème (1).

Remarque 2.

1. Le pas normal est calculé de manière approchée. Il doit simplement vérifier la région de confiance, appartenir à Ω et induire une réduction au moins aussi bonne que celle induite par un pas de Cauchy.
2. Le pas tangent est calculé de manière exacte. Cela fera l'objet d'une discussion dans la section 3.2.2.

Remarque 3. Lorsque le pas tangent est calculé de manière exacte, d'après les conditions KKT du premier ordre, il existe des multiplicateurs $y_k \in \mathbb{R}^m$ et $z_k \in \mathbb{R}^n$ ainsi qu'un sous gradient $g_{h,k} \in \partial h(x_k + v_k + u_k)$ tel que

$$g_k + \frac{1}{\nu_k} u_k + \frac{1}{\nu_k} v_k + g_{h,k} - J_k^\top y_k - z_k = 0. \tag{25}$$

Les multiplicateurs y_k sont associés à la contrainte $J_k u = 0$ tandis que les multiplicateurs z_k sont associés aux contraintes de bornes $x_k + v_k + u \in \Omega$.

Afin de résoudre le problème (22), nous utilisons la méthode présentée dans la section 3.1 pour les problèmes quadratiques de forme générale (18). Nous appliquons dans un premier temps la méthode du noyau, donnant le nouveau problème

$$\begin{aligned} \min_t \quad & g_k^\top Z t + \frac{1}{2\nu_k} \|Zt\|_2^2 + \frac{1}{\nu_k} v_k^\top Z t + h(x_k + v_k + Zt) \\ \text{tel que} \quad & x_k + v_k + u \in \Omega. \end{aligned} \tag{26}$$

Ensuite, nous résolvons (26) avec ENVELOPT pour $F(t) = Zt$. Cette approche permet de résoudre (1) avec divers régulariseurs (pourvu qu'ils soient convexes). Nous verrons dans la section 4.3.2 que cela permet de traiter des problèmes qui ne sont pas traités dans l'article de Curtis et al. [14].

3.2.2 Relaxation de la résolution du problème tangent

Ce développement théorique a fait l'objet d'une collaboration internationale avec Alberto De Marchi. Cette collaboration s'est notamment concrétisée par des réunions régulières en anglais, permettant

de discuter de l'avancement des travaux, ainsi que par la rédaction d'un rapport intermédiaire destiné à lui présenter l'état d'avancement de mes recherches. Les échanges avec Alberto De Marchi ont par ailleurs contribué à orienter la réflexion et à enrichir l'approche théorique développée.

Comme nous l'avons observé dans la Remarque 2, Curtis et al. [14] résolvent le problème (22) de manière exacte. Or, cette résolution peut s'avérer coûteuse pour le solveur, alors même qu'il n'est pas nécessaire de calculer une solution exacte à chaque itération. En effet, dans les premières itérations, il est souvent préférable de déterminer rapidement une direction de descente satisfaisante en résolvant le sous-problème de manière approchée, puis d'en améliorer progressivement la précision à mesure que les itérés se rapprochent d'un minimum local. Nous nous intéressons dans cette partie à la résolution du problème (22) de manière inexacte. En particulier, nous allons démontrer que sous certaines hypothèses, les résultats de convergence et de complexité restent inchangés. Nous simplifions le problème en enlevant les contraintes de bornes du problème initial (1), ce qui revient à résoudre le problème

$$\min_{x\in\mathbb{R}^n} f(x)+h(x) \quad \text{tel que} \quad c(x)=0. \tag{27}$$

L'ajout de bornes à l'analyse présentée ici pourra faire l'objet de travaux futurs. Pour adapter notre analyse, nous prendrons comme article de référence celui de Dai et al. [15], paru avant celui de Curtis et al. [14] et traitant du problème (1) sans les contraintes de bornes. Les deux articles utilisent la méthode SQP et l'algorithme est très sensiblement le même. L'objectif est de reprendre les résultats de Dai et al. [15] en les adaptant au cas où la résolution du problème tangent (22) n'est pas exacte. Les résultats issus de l'article et utilisés sans modification dans cette section sont présentés en Annexe F.

Hypothèses.

Commençons par rappeler les hypothèses nécessaires à l'analyse de convergence de [15].

Hypothèse 1. Soit $\mathcal{X}\subset\mathbb{R}^n$ un ensemble ouvert et convexe contenant les itérés $(x_k)_k\subset\mathbb{R}^n$ ainsi que les pas d'essai $(x_k+s_k)_k\subset\mathbb{R}^n$ générés par l'Algorithme 3. La fonction $f:\mathbb{R}^n\to\mathbb{R}$ est continûment différentiable et minorée sur $\mathcal{X}$, et sa fonction gradient $\nabla f:\mathbb{R}^n\to\mathbb{R}^n$ est Lipschitz-continue et bornée sur $\mathcal{X}$. De même, la fonction $c:\mathbb{R}^n\to\mathbb{R}^m$ est continûment différentiable et bornée sur $\mathcal{X}$, et sa jacobienne $\nabla c(x)^\top$ est Lipschitz-continue et bornée sur $\mathcal{X}$. Enfin, la fonction $h:\mathbb{R}^n\to\mathbb{R}_{\geq 0}$ est convexe, et son sous-différentiel $\partial h:\mathbb{R}^n\to\mathbb{R}^n$ est borné sur $\mathcal{X}$.

Sous l'hypothèse 1, il existe des constantes

$$(f_{\text{inf}},\kappa_{\nabla f},\kappa_{\partial h},\kappa_c,\kappa_{\nabla c},L_g,L_J)\in\mathbb{R}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}\times\mathbb{R}_{>0}$$

telles que, pour tout $x\in\mathcal{X}$, on ait

$$\begin{gathered} f(x)\geq f_{\text{inf}}, \qquad \|\nabla f(x)\|_2\leq\kappa_{\nabla f}, \qquad \|\partial h(x)\|_2\leq\kappa_{\partial h},\\ \|c(x)\|_2\leq\kappa_c, \qquad \|\nabla c(x)^\top\|_2\leq\kappa_{\nabla c}, \end{gathered} \tag{28}$$

et, pour tout $(x,\bar{x})\in\mathcal{X}\times\mathcal{X}$, on ait

$$\begin{aligned} \|\nabla f(x)-\nabla f(\bar{x})\|_2&\leq L_g\|x-\bar{x}\|_2,\\ \|\nabla c(x)^\top-\nabla c(\bar{x})^\top\|_2&\leq L_J\|x-\bar{x}\|_2. \end{aligned} \tag{29}$$

Hypothèse 2. Les plus petites valeurs singulières de la suite $\{J_k\}$ sont uniformément bornées inférieurement par une constante strictement positive. Autrement dit, il existe $\sigma_{\min}>0$ tel que, pour tout $k\in\mathbb{N}$,

$$\sigma_{\min}(J_k)\geq\sigma_{\min}.$$

Nous supposons ensuite deux hypothèses supplémentaires aux hypothèses 1 et 2.

Décroissance simple. Soit

$$q_k(u) = g_k^\top u + \frac{1}{2\nu_k}\|u\|_2^2 + h(x_k + v_k + u).$$

Nous supposons alors que la solution u_k de (22) respecte

$$q_k(u_k) \leq q_k(0) = h(x_k + v_k). \tag{30}$$

Décroissance suffisante. La solution du problème (22) n'étant plus exacte, nous introduisons un résidu dans les conditions KKT. En particulier en vertu de (25), en prenant en compte le fait qu'il n'y ait plus de contraintes de bornes et que $v_k^\top u = 0$ car $v_k \in \mathrm{Im}(J_k^\top)$ (voir [15, Eq. (2.2a)]), alors l'équation de stationnarité devient

$$g_k + \frac{u_k}{\nu_k} + g_{h,k} - J_k^\top y_k = r_k. \tag{31}$$

Nous supposons alors que la solution u_k de (22) respecte une décroissance suffisante telle que

$$\|r_k\| \leq \kappa_r \|g_k + g_{h,k} - J_k^\top y_k\|, \quad \kappa_r \in (0, \frac{1}{2}]. \tag{32}$$

Ce type de contrôle de résidu a déjà été appliqué aux méthodes SQP [8]. Nous pouvons également contrôler le résidu du sous-problème tangent en imposant que ce dernier doit rester une fraction contrôlée du terme de régularisation associé [27] au pas de sorte que

$$\|r_k\| \leq \kappa_r \frac{\|u_k\|}{\nu_k}. \tag{33}$$

Contribution 1. Nous introduisons deux nouvelles hypothèses portant sur la décroissance du modèle quadratique. Inspirées de la littérature, ces hypothèses permettent de contrôler le résidu tout en préservant les propriétés de convergence de l'algorithme, comme nous le démontrerons dans la suite de cette section.

Preuves

Nous nous intéressons à la démonstration de résultats de convergence et de complexité en considérant que l'algorithme ne termine pas en un nombre d'itérations fini. Nous commençons par introduire quatre résultats préliminaires. Le premier permet d'établir une inégalité de norme entre les conditions KKT et le résidu.

Lemme 1. Pour $a, b \in \mathbb{R}$, alors

$$\begin{aligned} &(a-b)^2 - (1/2a^2 - b^2) \\ &= 1/2a^2 - 2ab + 2b^2 \\ &= (1/\sqrt{2}a - \sqrt{2}b)^2 \geq 0 \end{aligned}$$

ce qui donne

$$(a-b)^2 \geq 1/2a^2 - b^2. \tag{34}$$

Nous appliquons (34) à (31) pour obtenir

$$\frac{\|u_k\|_2^2}{\nu_k^2} \geq \frac{1}{2}\|g_k + g_{h,k} - J_k^\top y_k\|_2^2 - \|r_k\|_2^2.$$

Le deuxième établit un résultat de continuité pour les sous-différentiels.

Lemme 2. ([38, Théorème 24.4]) Soit $h : \mathbb{R}^n \to \mathbb{R}$ une fonction propre, semi-continue inférieurement et convexe. Pour $(y_k)_k \to y$ et $(x_k)_k \to x$ tels que $y_k \in \partial h(x_k)$, alors $y \in \partial h(x)$.

Démonstration. Sachant que h est convexe, nous savons pour tout $z \in \mathbb{R}^n$,

$$h(z) \geq h(x_k) + y_k^\top (z - x_k)$$

Par semi-continuité inférieure de h et par continuité du produit scalaire, nous obtenons

$$h(x) \leq \liminf h(x_k) \leq h(z) - y_k^\top (z - x)$$

le fait que h est propre assure que $h(x) > -\infty$. Il vient, d'après

$$h(z) \geq h(x) + y^\top (z - x)$$

que $y \in \partial h(x)$. □

Le Théorème suivant fait un lien entre le noyau et l'espace des lignes d'une matrice M.

Théorème 3. ([42, Théorème 3C]) Soit $J \in \mathbb{R}^{m \times n}$. Le noyau $\ker(M)$ et l'image de $M^\top$ sont des sous-espaces orthogonaux supplémentaires de $\mathbb{R}^n$, c'est-à-dire

$$\ker(M) = \mathrm{Im}(M^\top)^\perp.$$

Enfin, le dernier Lemme donne des relations concernant les projections orthogonales sur l'espace des lignes d'une matrice.

Contribution 2. Ce lemme établit des relations entre les projections orthogonales sur l'espace des lignes d'une matrice et son noyau, et sera utilisé de manière centrale dans la vérification des hypothèses de la Remarque 5. En particulier, il permet de relier la condition d'arrêt d'Envelopt, exprimée dans l'espace réduit via $Z_k^\top$, au résidu r_k, rendant ainsi possible le contrôle de la décroissance suffisante (33) à partir des quantités effectivement calculées par le solveur.

Lemme 4. Soit Z une base orthonormale de $J \in \mathbb{R}^{m \times n}$ une matrice de rang plein. Alors $I - ZZ^\top$ et $J^\top (JJ^\top)^{-1} J$ sont des projections orthogonales sur $\mathrm{Im}(J^\top)$, et $\forall x \in \mathbb{R}^n$, $(I - ZZ^\top)x = J^\top (JJ^\top)^{-1} Jx$.

Démonstration. Soit Z une base orthonormale de $J \in \mathbb{R}^{m \times n}$ une matrice de rang plein.

1. Soit $x \in \mathbb{R}^n$, alors $ZZ^\top x = Zy \in \ker(J)$ car c'est une combinaisons linéaire de vecteurs d'une base du noyau de J. Puis pour $x \in \ker(J)$, il existe a tel que $x = Za$ de sorte que $ZZ^\top x = Za = x$. Enfin, pour $x \in \mathbb{R}^n$, $(ZZ^\top x)^\top (x - ZZ^\top x) = x^\top ZZ^\top x - x^\top ZZ^\top ZZ^\top x = 0$. Par conséquent, $ZZ^\top$ est la projection orthogonale sur $\ker(J)$ et d'après le Théorème 3, $I - ZZ^\top$ est la projection orthogonale sur $\ker(J)^\perp = \mathrm{Im}(J^\top)$.
2. De même, pour $x \in \mathbb{R}^n$, $J^\top (JJ^\top)^{-1} Jx \in \mathrm{Im}(J^\top)$, $J^\top (JJ^\top)^{-1} JJ^\top (JJ^\top)^{-1} Jx = J^\top (JJ^\top)^{-1} Jx$ et $\left(J^\top (JJ^\top)^{-1} J\right)^\top \left(I - J^\top (JJ^\top)^{-1} J\right) x = x^\top J^\top (JJ^\top)^{-1} Jx - x^\top J^\top (JJ^\top)^{-1} JJ^\top (JJ^\top)^{-1} Jx = 0$. Ainsi, $J^\top (JJ^\top)^{-1} J$ est la projection orthogonale sur $\mathrm{Im}(J^\top)$.
3. Par unicité de la projection orthogonale et l'hypothèse que J soit de rang plein,

$$(I - ZZ^\top)x = J^\top (JJ^\top)^{-1} J.$$ □

Dans la suite, nous noterons $\overline{\phi}$ la fonction de mérite shiftée telle que

$$\overline{\phi}_\tau(x) = \tau(f(x) - f_{\min} + h(x)) + \|c(x)\|$$

et $q_k(s, \tau)$ un modèle de la fonction de mérite tel que

$$q_k(s, \tau) = \tau \left(f_k + g_k^\top s + \frac{1}{2\nu_k} \|s\|_2^2 + h(x_k + s) \right) + \|c_k + J_k s\|_2.$$

Nous sommes à présent en mesure de démontrer que nous pouvons conserver les résultats de convergence et de complexité de Dai et al. [15] en résolvant de manière inexacte (22). Le lemme suivant donne une borne sur le dénominateur du paramètre de la fonction de mérite τ (voir Annexe E).

Contribution 3. La démonstration est adaptée de celle du Lemme 3.7 de Dai et al. [15] afin de tenir compte de la résolution inexacte du problème tangent ainsi que des hypothèses de décroissance simple et suffisante. La modification concerne principalement les trois premières inégalités.

Lemme 5. ([15, Lemme 3.7]) Pour tout $k \in \mathbb{N}$, il s'ensuit que

$$g_k^\top s_k + \frac{\|s_k\|_2^2}{2\nu_k} + h(x_k + s_k) - h(x_k) \leq (\kappa_{\nabla f} + \kappa_{\partial h})\|v_k\|_2 + \frac{\|v_k\|_2^2}{2\nu_k}$$

Démonstration. Soit $k \in \mathbb{N}$, alors

$$\begin{aligned}
& g_k^\top s_k + \frac{\|s_k\|_2^2}{2\nu_k} + h(x_k + s_k) - h(x_k) \\
(i) \quad & = g_k^\top s_k + h(x_k + s_k) - h(x_k) + \frac{\|u_k\|^2}{2\nu_k} + \frac{\|v_k\|^2}{2\nu_k} \\
(ii) \quad & \leq (g_k + g_{h,k})^\top s_k + \frac{\|u_k\|^2}{2\nu_k} + \frac{\|v_k\|^2}{2\nu_k} \\
(iii) \quad & \leq (\kappa_{\nabla f} + \kappa_{\partial h})\|v_k\|_2 + \frac{\|v_k\|_2^2}{2\nu_k} + (g_k + g_{h,k})^\top u_k + \frac{\|u_k\|^2}{2\nu_k} \\
(iv) \quad & = (\kappa_{\nabla f} + \kappa_{\partial h})\|v_k\|_2 + \frac{\|v_k\|_2^2}{2\nu_k} + r_k^\top u_k - \frac{\|u_k\|^2}{2\nu_k} \\
(v) \quad & \leq (\kappa_{\nabla f} + \kappa_{\partial h})\|v_k\|_2 + \frac{\|v_k\|_2^2}{2\nu_k} + \kappa_r \frac{\|u_k\|^2}{\nu_k} - \frac{\|u_k\|^2}{2\nu_k} \\
(vi) \quad & \leq (\kappa_{\nabla f} + \kappa_{\partial h})\|v_k\|_2 + \frac{\|v_k\|_2^2}{2\nu_k}
\end{aligned}$$

où (i) vient de la relation $v_k^\top u_k = 0$ (voir [15, Eq. (2.2a)]), (ii) utilise l'inégalité de convexité $h(x_k) \geq h(x_k + s_k) + (g_{h,k})^\top(-s_k)$, $g_{h,k} \in \partial h(x_k + s_k)$, (iii) découle de Cauchy-Schwarz et de l'hypothèse 1, (iv) s'obtient en multipliant (31) par $u_k^\top$, (v) provient de la décroissance suffisante (33) et (vi) d'après l'hypothèse $\kappa_r \in (0, \frac{1}{2}]$. □

Avant d'énoncer le théorème donnant un résultat de complexité pour l'Algorithme 3, nous discutons de la métrique de stationnarité utilisée pour cet algorithme. Dai et al. [15] utilisent

$$\chi_k = \max\{\|g_k + \tilde{g}_{h,k} - J_k^\top y_k\|_2, \|c_k\|_2\} \tag{35}$$

comme mesure de stationnarité pour le problème (1), où $\tilde{g}_{h,k} \in \partial h(x_k + v_k + u_k)$. Cependant, nous n'avons numériquement pas accès à un tel sous gradient. Notons que dans le cadre des articles [14] et [15], les auteurs se restreignent à $h = \|\cdot\|_1$. Par conséquent, après la transformation décrite dans l'Annexe B, le problème devient lisse. Cela permet de trouver facilement un sous gradient : celui-ci est unique et coïncide avec le gradient.

Nous nous appuyons sur les informations fournies par ENVELOPT, le solveur utilisé pour résoudre (22). L'algorithme, en résolvant une série de sous-problèmes (17), détermine u_k et w_k tels que :

$$\begin{cases}
J_k u_k = 0, \\
w_k = \operatorname{prox}_{\mu_k h}(u_k + \mu_k \hat{y}_k) \\
g_{h,k} = \hat{y}_k + \frac{u_k - w_k}{\mu_k} \in \partial h(x_k + v_k + w_k)\ ^{5}, \\
\|u_k - w_k\| \leq \epsilon_{Env,k}, \\
\|Z_k^\top \left(g_k + \frac{1}{\nu_k} u_k + g_{h,k}\right)\| \leq \epsilon_{Env,k},
\end{cases} \tag{36}$$

où $\hat{y}_k$ sont les multiplicateurs associés aux sous-problèmes résolus par ENVELOPT. Si nous permettons à ENVELOPT de ne pas résoudre de manière exacte le problème tangent, alors nous avons un résidu

5. $\operatorname{prox}_h(x) = \arg\min_y\{\frac{1}{2}\|x - y\|_2^2 + h(y)\} \Longrightarrow 0 \in \partial h(\operatorname{prox}_h(x)) - x + \operatorname{prox}_h(x)$.

pour ce sous-problème tel que

$$\begin{cases} u_k - w_k = \rho_k, \\ g_k + \frac{1}{\nu_k} u_k + g_{h,k} - J_k^\top y_k = r_k. \end{cases}$$

Lorsque nous multiplions la seconde équation par $Z_k^\top$ et en rappelant que $Z_k^\top$ est calculé de sorte que $J_k Z_k = 0$, nous obtenons la condition de stationnarité (36). Il convient de noter ici que $g_{h,k} \in \partial h(x_k + v_k + w_k)$ auquel nous avons accès grâce à ENVELOPT. Nous introduisons alors notre nouvelle mesure de stationnarité

$$\chi_k = \max\{\|g_k + g_{h,k} - J_k^\top y_k\|_2, \|c_k\|_2\}, \tag{37}$$

où $g_{h,k} \in \partial h(x_k + v_k + w_k)$. La motivation de ce changement est qu'ENVELOPT nous donne accès à un sous gradient de $\partial h(x_k+v_k+w_k)$ et non de $\partial h(x_k+v_k+u_k)$. Cette nouvelle mesure est donc construite à partir d'un sous-gradient effectivement disponible lors de la résolution numérique du problème tangent.

Remarque 4. Si l'on souhaite obtenir un point stationnaire, il faut imposer que le résidu tende vers 0. Pour ce faire, nous imposons $\epsilon_{Env,k} \to 0$.

Nous démontrons à présent que la résolution du problème tangent de manière inexacte conserve le résultat de complexité énoncé dans [15].

Contribution 4. La démonstration est adaptée de celle du Théorème 3.15 de Dai et al. [15] afin de tenir compte de la résolution inexacte du problème tangent par l'introduction d'un résidu. Ce résidu est ensuite contrôlé grâce à l'hypothèse de décroissance suffisante, ce qui permet d'obtenir un résultat similaire à celui du Théorème 3.15 de Dai et al. [15].

Théorème 6. ([15, Théorème 3.15]) Supposons que les hypothèses 1 et 2 soient vérifiées, et soit $\epsilon \in \mathbb{R}_{>0}$ donné. Si $\{k_1, k_2\} \subset \mathbb{N}$ sont deux itérations telles que $k_1 < k_2$, et que $k \in \mathcal{S}$ et $\chi_k > \epsilon$ pour toutes les itérations $k_1 \leq k < k_2$, alors

$$k_2 - k_1 \leq \left\lfloor \frac{\tau_0\big(f(x_0) - f_{\inf} + r(x_0)\big) + \|c(x_0)\|_2}{\kappa_\Phi \epsilon^2} \right\rfloor, \tag{38}$$

où

$$\kappa_\Phi := \eta \min\left\{ \left(\frac{1}{2} - \kappa_r^2\right) \tau_{\min}\nu_{\min}, \frac{\sigma_c \sigma_{\min}^2}{2\kappa_c(1+\kappa_{\nabla c}^2)}, \frac{\sigma_c \sigma_{\min}^2 \kappa_v \nu_{\min}}{2\kappa_c} \right\}.$$

De plus, le nombre maximal d'itérations avant que $\chi_k \leq \epsilon$ pour une certaine itération $k \in \mathbb{N}$ est donné par

$$\left(\max\left\{ 0, \left\lceil \frac{\log\left(\frac{\tau_{\min}}{\nu_0(\tau_{\min}L_g + L_J)}\right)}{\log(\xi)} \right\rceil \right\} + 1 \right) \left\lfloor \frac{\tau_0\big(f(x_0) - f_{\inf} + r(x_0)\big) + \|c(x_0)\|_2}{\kappa_\Phi \epsilon^2} \right\rfloor. \tag{39}$$

Démonstration. Soient $\{k_1, k_2\} \subset \mathbb{N}$ tels que définis dans l'énoncé de ce théorème. Alors, en vertu du Lemme 11, des parties (i)–(ii) du Lemme 14, de la ligne 18 de l'Algorithme 3, du Lemme 9, du Lemme 12 et du Lemme 13, les inégalités suivantes sont vérifiées pour tout $k_1 \leq k < k_2$:

$$\begin{aligned} \bar{\phi}_{\tau_k}(x_k) - \bar{\phi}_{\tau_{k+1}}(x_{k+1}) &\geq \bar{\phi}_{\tau_k}(x_k) - \bar{\phi}_{\tau_k}(x_{k+1}) \\ &= \phi_{\tau_k}(x_k) - \phi_{\tau_k}(x_{k+1}) \\ &\geq \eta \Delta q_k(s_k, \tau_k) \\ &\geq \eta \frac{\tau_k}{2\nu_k}\|s_k\|_2^2 + \eta\sigma_c\big(\|c_k\|_2 - \|c_k + J_k v_k\|_2\big) \\ &\geq \eta \frac{1}{2}\tau_{\min}\nu_{\min}\left(\frac{\|s_k\|_2}{\nu_k}\right)^2 + \eta\sigma_c\big(\|c_k\|_2 - \|c_k + J_k v_k\|_2\big). \end{aligned}$$

En combinant cette inégalité avec $s_k = v_k + u_k$ et $v_k^\top u_k = 0$ pour tout $k \in \mathbb{N}$, le Lemme 10, le Lemme 13, le Lemme 1, la condition de décroissance suffisante (32), le fait que la suite $(\nu_k)_k$ est monotone décroissante et l'hypothèse 2, on obtient, pour tout $k_1 \leq k < k_2$,

$$\begin{aligned}
\bar{\phi}_{\tau_k}(x_k) - \bar{\phi}_{\tau_{k+1}}(x_{k+1}) \geq & \eta \frac{1}{2} \tau_{\min} \nu_{\min} \left(\frac{\|u_k\|_2}{\nu_k} \right)^2 \\
& + \eta \sigma_c \frac{1}{2\kappa_c} \|J_k^\top c_k\|_2^2 \min \left\{ \frac{1}{1 + \kappa_{\nabla c}^2}, \kappa_v \nu_{\min} \right\} \\
\geq & \eta \tau_{\min} \nu_{\min} \left(\frac{1}{2} \|g_k + g_{h,k} - J_k^\top y_k\|_2^2 - \|r_k\|_2^2 \right) \\
& + \eta \sigma_c \frac{\sigma_{\min}^2}{2\kappa_c} \|c_k\|_2^2 \min \left\{ \frac{1}{1 + \kappa_{\nabla c}^2}, \kappa_v \nu_{\min} \right\} \\
\geq & \eta \left(\frac{1}{2} - \kappa_r^2 \right) \tau_{\min} \nu_{\min} \|g_k + g_{h,k} - J_k^\top y_k\|_2^2 \\
& + \eta \sigma_c \frac{\sigma_{\min}^2}{2\kappa_c} \|c_k\|_2^2 \min \left\{ \frac{1}{1 + \kappa_{\nabla c}^2}, \kappa_v \nu_{\min} \right\} \\
\geq & \kappa_\Phi \chi_k^2,
\end{aligned}$$

où κ_Φ est défini dans l'énoncé du théorème. Nous pouvons remarquer que comme $\kappa_r \leq \frac{1}{2}$, la quantité $(\frac{1}{2} - \kappa_r^2)$ est bien positive. En utilisant cette inégalité, la partie (iii) du Lemme 14 et la non-négativité de $\bar{\phi}_\tau$ pour tout $\tau \in \mathbb{R}_{>0}$, on obtient

$$\begin{aligned}
\bar{\phi}_{\tau_0}(x_0) & \geq \bar{\phi}_{\tau_{k_1}}(x_{k_1}) \\
& \geq \bar{\phi}_{\tau_{k_1}}(x_{k_1}) - \bar{\phi}_{\tau_{k_2}}(x_{k_2}) \\
& \geq \sum_{k=k_1}^{k_2-1} \left(\bar{\phi}_{\tau_k}(x_k) - \bar{\phi}_{\tau_{k+1}}(x_{k+1}) \right) \\
& \geq \sum_{k=k_1}^{k_2-1} \kappa_\Phi \chi_k^2.
\end{aligned}$$

Cette inégalité peut être combinée avec le fait que $\chi_k > \epsilon$ pour toutes les itérations $k_1 \leq k < k_2$ afin d'obtenir

$$\bar{\phi}_{\tau_0}(x_0) \geq (k_2 - k_1) \kappa_\Phi \epsilon^2,$$

d'où découle (38). Enfin, le dernier résultat du théorème, à savoir la borne supérieure sur le nombre maximal d'itérations avant que $\chi_k \leq \epsilon$, découle du résultat précédent et du fait que le nombre maximal d'itérations infructueuses est borné comme indiqué dans le Lemme 13. □

La preuve du Théorème 6 peut être ajustée en utilisant la décroissance suffisante (33) en changeant $\left(\frac{1}{2} - \kappa_r^2\right) \tau_{\min} \nu_{\min}$ par $\frac{1}{(1+\kappa_r)^2} \tau_{\min} \nu_{\min}$. En effet en notant $R_k = g_k + g_{h,k} - J_k^\top y_k$, alors $R_k = r_k - \frac{u_k}{\nu_k}$, ce qui implique que

$$\begin{aligned}
\|R_k\|_2 & \leq \|r_k\|_2 + \frac{\|u_k\|_2}{\nu_k} \\
& \leq (1 + \kappa_r) \frac{\|u_k\|_2}{\nu_k} \\
\implies \left(\frac{\|u_k\|_2}{\nu_k} \right)^2 & \geq \frac{1}{(1 + \kappa_r)^2} \|R_k\|_2.
\end{aligned}$$

En pratique nous vérifions que la décroissance suffisante (32) ou (33) est respectée. Le résultat suivant permet de borner les multiplicateurs de Lagrange.

Contribution 5. La démonstration est adaptée de celle du Lemme 3.16 de Dai et al. [15]. Notre nouvelle hypothèse de décroissance simple permet d'obtenir une borne sur la norme de u_k et nous tenons compte de la résolution inexacte du problème tangent par l'introduction d'un résidu dans l'expression des multiplicateurs.

Lemme 7. ([15, Lemme 3.16]) La suite des estimateurs des multiplicateurs de Lagrange $(y_k)_k$ est bornée.

Démonstration. A partir de la décroissance simple (30), l'inégalité de convexité, l'inégalité de Cauchy-Schwarz et l'inégalité triangulaire nous savons que

$$\begin{aligned}\frac{1}{\nu_k}\|u_k\|_2^2 &\leq -g_k^\top u_k + h(x_k+v_k) - h(x_k+v_k+u_k)\\ &\leq -(g_k+g_{h,k})^\top u_k\\ &\leq \|g_k+g_{h,k}\|_2\|u_k\|_2\\ &\leq \big(\|g_k\|_2+\|g_{h,k}\|_2\big)\|u_k\|_2,\end{aligned}$$

qui, combiné à l'hypothèse 1, montre que

$$\frac{1}{\nu_k}\|u_k\|_2 \leq \kappa_{\nabla f}+\kappa_{\partial h}. \tag{40}$$

De plus, d'après la décroissance suffisante (32) et (40) nous avons

$$\begin{aligned}\|r_k\|_2 &\leq \kappa_r\left(\|r_k\|_2+\frac{\|u_k\|_2}{\nu_k}\right)\\ &\leq \kappa_r\|r_k\|+\kappa_r(\kappa_{\nabla f}+\kappa_{\partial h}),\end{aligned}$$

ce qui donne

$$\|r_k\|_2 \leq \frac{\kappa_r}{1-\kappa_r}(\kappa_{\nabla f}+\kappa_{\partial h}). \tag{41}$$

Il découle de (31) et l'hypothèse 2 que

$$J_k^\top y_k = g_k+\frac{1}{\nu_k}u_k+g_{h,k}-r_k \quad\Longleftrightarrow\quad y_k=(J_kJ_k^\top)^{-1}J_k\left(g_k+\frac{1}{\nu_k}u_k+g_{h,k}-r_k\right). \tag{42}$$

En combinant (42), l'hypothèse 2, l'inégalité triangulaire, et les relations (41) et (40), il vient que

$$\begin{aligned}\|y_k\|_2 &\leq \frac{1}{\sigma_{\min}}\left\|g_k+\frac{1}{\nu_k}u_k+g_{h,k}\right\|_2+\frac{1}{\sigma_{\min}}\|r_k\|_2\\ &\leq \frac{1}{\sigma_{\min}}\left(\kappa_{\nabla f}+\kappa_{\partial h}+\frac{1}{\nu_k}\|u_k\|_2\right)+\frac{1}{\sigma_{\min}}\|r_k\|_2\\ &\leq \frac{2}{\sigma_{\min}}\big(\kappa_{\nabla f}+\kappa_{\partial h}\big)+\frac{\kappa_r(\kappa_{\nabla f}+\kappa_{\partial h})}{\sigma_{\min}(1-\kappa_r)}\\ \Longrightarrow \|y_k\|_2 &\leq \frac{1}{\sigma_{\min}}\left(2+\frac{\kappa_r}{1-\kappa_r}\right)(\kappa_{\nabla f}+\kappa_{\partial h}).\end{aligned}$$

Comme ce résultat est valable pour tout $k\in\mathbb{N}$, cela termine la preuve. □

A l'instar du Théorème 6, la preuve du Lemme 7 peut être ajustée en utilisant la décroissance suffisante (33) en changeant $\frac{1}{\sigma_{\min}}\left(2+\frac{\kappa_r}{1-\kappa_r}\right)(\kappa_{\nabla f}+\kappa_{\partial h})$ par $\frac{1}{\sigma_{\min}}(2+\kappa_r)(\kappa_{\nabla f}+\kappa_{\partial h})$. En effet, $\|r_k\| \leq \kappa_r\frac{\|u_k\|}{\nu_k} \leq \kappa_r(\kappa_{\nabla f}+\kappa_{\partial h})$, donc $\frac{1}{\sigma_{\min}}\left(\kappa_{\nabla f}+\kappa_{\partial h}+\frac{1}{\nu_k}\|u_k\|_2\right)+\frac{1}{\sigma_{\min}}\|r_k\|_2 \leq \frac{1}{\sigma_{\min}}(2+\kappa_r)(\kappa_{\nabla f}+\kappa_{\partial h})$.

Nous pouvons à présent montrer que les points d'adhérence de la suite de points générée par l'Algorithme 3 sont des points vérifiant la condition KKT du premier ordre.

Théorème 8. ([15, Théorème 3.17]) Supposons que les hypothèses 1 et 2 soient vérifiées. Tout point d'adhérence $x_\star$ de la suite $(x_k)_k$ qui est aussi limite d'une sous-suite $(x_k)_{k\in\mathcal{K}_1}$ telle que $\chi_k \to 0$ est un point KKT du premier ordre du problème (27).

Démonstration. Soit $x_\star$ un point d'adhérence de $(x_k)_k$. Le Théorème 6 permet de conclure qu'il existe un sous ensemble $\mathcal{K}_1$ telle que

$$0 = \lim_{k\in\mathcal{K}_1} \chi_k = \lim_{k\in\mathcal{K}_1} \max\left\{\|g_k + g_{h,k} - J_k^\top y_k\|_2, \|c_k\|_2\right\}. \tag{43}$$

Nous savons également que $x_\star$ est toujours un point d'adhérence pour la suite $(x_k)_{k\in\mathcal{K}_1}$, c'est-à-dire qu'il existe un sous ensemble $\mathcal{K}_2 \subseteq \mathcal{K}_1$ tel que $(x_k)_{k\in\mathcal{K}_2} \to x_\star$. Le Lemme 7 permet également d'affirmer qu'il existe un vecteur $y_\star \in \mathbb{R}^m$ et une sous-suite $\mathcal{K}_3 \subseteq \mathcal{K}_2$ tels que $(y_k)_{k\in\mathcal{K}_3} \to y_\star$. Par continuité de J et c, il vient $(c(x_k))_{k\in\mathcal{K}_3} \to 0 = c(x_\star)$, ce qui implique $(J(x_k)^\top c(x_k))_{k\in\mathcal{K}_3} \to J(x_\star)^\top c(x_\star) = 0$. En vertu de (21) et du Lemme 13, nous obtenons alors $(v_k)_{k\in\mathcal{K}_3} \to 0$.

De plus, $R_k = g_k + g_{h,k} - J_k^\top y_k \Leftrightarrow g_{h,k} = J_k^\top y_k + R_k - g_k$. D'après (43) et ce qui précède, $\lim_{\mathcal{K}_3} g_{h,k} = \lim_{\mathcal{K}_3}(J_k^\top y_k + R_k - g_k) = J^\top(x_\star)y_\star - g(x_\star)$. Autrement dit, il existe $g_{h,\star}$ tel que $\lim_{\mathcal{K}_3} g_{h,k} = g_{h,\star}$.

Enfin, en combinant la Remarque 4 et le système d'équations (36), nous obtenons $\|u_k - w_k\| \to 0$. D'après le [14, Lemme 5.23], $\lim_{k\in\mathcal{S}} \|s_k\| = 0$ où $\mathcal{S}$ est l'ensemble des indices des pas acceptés. Ainsi, $\|s_k\|^2 = \|v_k\|^2 + \|u_k\|^2 \to 0$ et étant donné que $v_k \to 0$, nous avons $u_k \to 0$. Comme $\|u_k - w_k\| \to 0$, il vient que $w_k \to 0$. Finalement $x_k + v_k + w_k \to x_\star$, ce qui, puisque $g_{h,k} \in \partial h(x_k + v_k + w_k)$, implique $g_{h,\star} \in \partial h(x_\star)$ en vertu du Lemme 2. Enfin, il découle de $g(x_\star) + g_{h,\star} - J(x_\star)^\top y_\star = 0$ et de $c(x_\star) = 0$ que $x_\star$ est un point KKT du premier ordre du problème (27). □

Contribution 6. La démonstration est adaptée de celle du Théorème 3.17 de Dai et al. [15] afin de tenir compte du fait que $g_{h,k} \in \partial h(x_k + v_k + w_k)$, et non $g_{h,k} \in \partial h(x_k + v_k + u_k)$. De plus, la preuve originale du Théorème 8, considère directement une sous-suite sur laquelle le critère χ_k converge vers zéro, tout en conservant la convergence vers le point d'adhérence $x_\star$. Dans notre cadre, afin de rendre explicite ce passage, nous supposons que $x_\star$ demeure un point d'adhérence de la sous-suite $\{x_k\}_{k\in\mathcal{K}_1}$ fournie par le Théorème 6, pour laquelle $\chi_k \to 0$. En effet, le Théorème 6 garantit l'existence d'une sous-suite d'indices $\mathcal{K}_1$ telle que $\chi_k \to 0$, tandis que le fait que $x_\star$ soit un point d'adhérence de la suite x_k garantit a priori l'existence d'une sous-suite convergeant vers $x_\star$. Ces deux sous-suites ne coïncident pas nécessairement et leur intersection n'est pas nécessairement non vide. L'hypothèse supplémentaire permet donc de s'assurer qu'il existe une sous-suite $\mathcal{K}_2 \subseteq \mathcal{K}_1$ telle que

$$x_k \xrightarrow[k\in\mathcal{K}_2]{} x_\star \qquad \text{et} \qquad \chi_k \xrightarrow[k\in\mathcal{K}_2]{} 0,$$

ce qui est nécessaire pour poursuivre la preuve de la stationnarité de $x_\star$.

Contribution 7. L'hypothèse 1 suppose que pour le h choisit, il existe une constante $\kappa_{\partial h}$ tel que

$$\forall x \in \mathbb{R}^n,\ \|\partial h(x)\| \leq \kappa_{\partial h}. \tag{44}$$

Cette hypothèse est très restrictive car elle écarte notamment les fonctions indicatrices. En effet, en notant

$$\mathbf{1}_\Omega(x) = \begin{cases} 0, & x \in \Omega \\ +\infty, & x \notin \Omega \end{cases}$$

alors la sous-différentielle est le cône normal tel que $\partial \mathbf{1}_\Omega(x) = N_\Omega(x)$. Si $x \in \partial\Omega$, alors le cône normal peut être non borné. Dans les démonstrations du Lemme 5 et du Lemme 7, nous utilisons uniquement des sous-gradients $g_{h,k} \in \partial h(x_k + s_k)$, dont le choix est laissé libre. Nous pouvons ainsi, à chaque itération, choisir un sous-gradient de norme minimale. Cela permet de relâcher l'hypothèse (44) en supposant plutôt que

$$\sup_k \min\{\|g_{h,k}\| \mid g_{h,k} \in \partial h(x_k + s_k)\} \leq \kappa_{\partial h}.$$

Sous couvert d'une décroissance simple et d'une décroissance suffisante, nous avons donc démontré que la résolution inexacte de (22) n'affecte pas le Lemme 5 qui donne des informations importantes sur le comportement du paramètre de mérite (voir Annexe E). Par exemple, lorsque $v_k = 0$ (ce qui signifie que l'on a atteint la faisabilité si $c_k = 0$), alors (E2) donne $\tau_{k,\text{trial}} = \infty$ en vertu du Lemme 5. Ce lemme est très utilisé dans d'autres démonstrations de l'article [15] et il était donc crucial de maintenir ce résultat. Fort de ce résultat, nous sommes alors capables de conserver le résultat de complexité établi au Théorème 6 en ajustant la mesure de stationnarité (voir (37)). Ce théorème, couplé à la bornitude des multiplicateurs de Lagrange obtenue grâce au Lemme 7 permet de maintenir le résultat de convergence de l'Algorithme (3) à l'aide du Théorème 8. Enfin, nous avons été capables d'améliorer l'hypothèse de borne sur le sous-différentiel de h.

Remarque 5. Le changement de métrique de stationnarité a été motivé par l'utilisation du solveur ENVELOPT. Cependant de manière plus générale, tout solveur résolvant (22) de sorte que les contraintes $J_k u = 0$ soient satisfaites de manière exacte, et fournissant les quantités $(r_k, g_{h,k}, w_k)$ telles que

$$\begin{cases} \|r_k\| \leq \kappa_r \frac{\|u_k\|}{\nu_k} \quad \text{ou} \quad \|r_k\| \leq \kappa_r \|g_k + g_{h,k} - J_k^\top y_k\|, \\ g_{h,k} \in \partial h(x_k + v_k + w_k), \\ \|u_k - w_k\| \leq \epsilon_k, \ \epsilon_k \to 0. \end{cases}$$

satisfait les conditions requises dans les preuves précédentes. Ainsi, les résultats établis restent valables et un tel solveur peut également être utilisé.

4 Etude numérique

4.1 Développements logiciels et contributions open source

La mise en œuvre des méthodes présentées dans ce rapport s'est appuyée sur plusieurs packages Julia [4] développés au sein de l'organisation open source JuliaSmoothOptimizers. Dans le cadre de ce stage, j'ai contribué à ces packages par l'intermédiaire de plusieurs *pull requests*, notamment pour l'ajout de fonctionnalités et l'amélioration de certaines implémentations. J'ai également ouvert des *issues* afin de signaler et de corriger certains comportements inattendus ou problèmes rencontrés lors de leur utilisation.

En complément de ces contributions, j'ai développé un solveur dédié à la résolution du problème (1) ($\simeq$ 2400 lignes de code). Celui-ci s'appuie sur les fonctionnalités fournies par les différents packages de l'organisation et implémente la méthode présentée dans la section 3. Cette approche permet notamment de réutiliser le code et les interfaces déjà existantes, tout en développant les composantes spécifiques nécessaires à la méthode proposée. Les expériences numériques présentées dans les sections suivantes s'appuient sur ces développements.

4.2 Résolution de problèmes quadratiques non lisses

4.2.1 Détails d'implémentation

Création du problème quadratique. Afin de construire un problème quadratique de la forme (18), nous sélectionnons tout d'abord un problème comportant des contraintes d'égalité dans le package `OptimizationProblems` [30] [6], issu de la bibliothèque JuliaSmoothOptimizers (JSO). À partir d'un point initial x, nous construisons ensuite le modèle quadratique du second ordre de la fonction objectif à l'aide du package `QuadraticModels` [26] (également issu de JSO) [7], ce qui conduit au problème

$$\varphi(s) = \nabla f(x)^\top s + \frac{1}{2} s^\top H s, \qquad \text{sous les contraintes d'égalité} \qquad Js = 0, \tag{45}$$

6. JuliaSmoothOptimizers/OptimizationProblems.jl
7. JuliaSmoothOptimizers/QuadraticModels.jl

où $J \in \mathbb{R}^{m\times n}$ désigne la jacobienne des contraintes d'égalité. Nous choisissons ensuite une fonction convexe non lisse ψ provenant du package `ProximalOperators` [8], ce qui permet d'obtenir le problème d'optimisation non lisse

$$\min_s \ \varphi(s) + \psi(s) \qquad \text{sous la contrainte} \qquad Js = 0. \tag{46}$$

Certains problèmes quadratiques construits de cette manière peuvent être non bornés. Lorsque c'est le cas, nous nous autorisons à modifier le problème en remplaçant la hessienne H par $H + \sigma I$, où σ est un paramètre déterminé automatiquement. La procédure adoptée est la suivante :

1. Choisir un facteur $\tau_\sigma > 1$, une valeur maximale $\sigma_{\max} > 0$ ainsi qu'une valeur initiale $\sigma > 0$.
2. Résoudre le problème (46) avec ENVELOPT en utilisant IPOPT comme solveur interne.
3. Si IPOPT détecte que le problème est non borné, remplacer H par $H + \sigma I$.
4. Mettre à jour σ selon $\sigma \leftarrow \tau_\sigma \sigma$.
5. Résoudre le nouveau problème avec ENVELOPT et IPOPT.
6. Si le problème est toujours non borné, répéter la procédure ; sinon, conserver la valeur finale de σ et arrêter.
7. Interrompre également la procédure lorsque $\sigma > \sigma_{\max}$.

En pratique, nous choisissons

$$\sigma_{\max} = 10^4, \qquad \tau_\sigma = 10, \qquad \sigma_0 = 10^{-3}.$$

Si, à l'issue de cette procédure, le problème demeure non borné, il est écarté des expériences numériques. Dans le cas contraire, la valeur finale de σ est utilisée pour construire le nouveau problème non lisse

$$\min_s \ \nabla f(x)^\top s + \frac{1}{2} s^\top (H + \sigma I) s, \qquad \text{sous la contrainte} \qquad Js = 0.$$

Afin de comparer équitablement les différents solveurs (ENVELOPT et la méthode du Lagrangien augmenté), il est important d'utiliser la même valeur de σ pour l'ensemble des expériences numériques.

Calcul d'une base du noyau. ENVELOPT nécessite une base Z du noyau de la matrice J. Pour la construire, nous utilisons le Théorème 3. En calculant la décomposition QR de $J^\top$ et en notant r le rang de $J^\top$, il est possible d'extraire une base de $\mathrm{Im}(J^\top)^\perp$. En effet, les r premières colonnes de Q engendrent $\mathrm{Im}(J^\top)$, tandis que les $n - r$ dernières colonnes engendrent son orthogonal [9]. La décomposition QR est réalisée au moyen du module `QRMumps` [6] [10], permettant de traiter efficacement les matrices creuses.

Contraintes d'égalité de la forme $Ax = b$. Nous avons jusqu'à présent considéré le cas où les contraintes d'égalité s'écrivent sous la forme $Ax = 0$. Cependant, certains problèmes issus de `OptimizationProblems` conduisent, après la construction du modèle quadratique avec `QuadraticModels`, à des contraintes de la forme $Ax = b$. Pour traiter ce cas dans ENVELOPT, nous décomposons la variable sous la forme $x = x_h + x_p$, où x_p est une solution particulière de l'équation $Ax = b$, tandis que x_h appartient au noyau de A, c'est-à-dire $Ax_h = 0$. On obtient alors $Ax = A(x_h + x_p) = Ax_p = b$, ce qui garantit que x satisfait les contraintes d'égalité. Il suffit alors de remplacer l'application $F(t) = Zt$ par $F(t) = Zt + x_p$, ce qui ne constitue qu'une modification mineure de l'algorithme. Enfin, la solution optimale est désormais donnée par $x^\star = Zt^\star + x_p$.

4.2.2 Résultats numériques

Nous résolvons les problèmes quadratiques créés avec le processus détaillé dans la partie 4.2.1 en comparant deux méthodes, l'une étant ENVELOPT détaillé en partie 3.1 et l'autre étant un Lagrangien

8. JuliaFirstOrder/ProximalOperators.jl
9. La matrice R est construite de telle sorte que les premières valeurs diagonales sont non nulles, puis nulles.
10. JuliaSmoothOptimizers/QRMumps.jl

augmenté adapté au problème non lisse. Cette méthode repose sur le fait d'ajouter le terme non lisse h au Lagrangien augmenté du problème, puis de résoudre les sous-problèmes avec un solveur traitant les problèmes non lisses. Cette méthode est détaillée dans la partie D. Nous sélectionnons des problèmes avec divers nombres de variables et de contraintes. L'idée étant, à terme, de résoudre des problèmes avec d'autres régulariseurs que $\|\cdot\|_1$ contrairement à [14], nous présentons ici des résultats numériques préliminaires obtenus avec différents régulariseurs. Nous commençons par un problème simple, avec $\psi(x) = \lambda\|x\|_1$ et $\lambda = 1.5$. Le problème `hs27` comporte trois variables et une contrainte. Le point initial pour créer le problème quadratique est celui proposé par défaut dans `OptimizationProblems`. Les résultats sont répertoriés dans le tableau 1.

Tableau 1 – Résultats numériques pour un problème quadratique obtenu avec `hs27` et $\psi(x) = \lambda\|x\|_1$.

Méthode	obj	iter	subiter	cviol	time (s)	status	primal_feas	dual_feas
Env-Tron	4.32	8	194	1.78e-15	8.82e-5	first_order	0.0	4.91e-6
Env-Trunk	4.32	4	257	1.78e-15	1.37e-3	first_order	0.0	6.26e-8
Env-IPOPT	4.32	4	14	8.88e-16	4.15e-3	first_order	0.0	6.69e-6
Env-MadNLP	4.32	4	16	8.88e-16	2.11e-3	first_order	0.0	1.47e-8
Lagrangien-R2	3.86	8	97	7.67e-10	3.67e-5	first_order	7.67e-10	8.11e-6
Lagrangien-R2N	3.86	8	4	4.74e-9	7.78e-5	first_order	4.74e-9	8.99e-6

Dans ce tableau, la colonne "Méthode" renseigne la méthode ainsi que le sous-solveur utilisé. Par exemple, *Env-Tron* signifie que l'on a utilisé la méthode 3.1 avec le sous-solveur TRON et *Lagrangien-R2* signifie que l'on a utilisé la méthode D avec le sous-solveur R2. La colonne "obj" correspond à la valeur de la fonction objectif après que l'algorithme ait terminé. Nous ferons remarquer que "terminer" ne signifie pas converger car l'algorithme peut s'arrêter lorsqu'un nombre d'itérations ou une durée maximale est atteint ou pour d'autres critères selon les algorithmes utilisés. Les colonnes "iter" et "subiter" représentent respectivement le nombre d'itérations de l'algorithme et le nombre d'itérations du sous-solveur. La quantité "cviol" est la violation des contraintes $\|c(x^\star)\|$. Nous répertorions également le temps en seconde qu'a mis l'algorithme à converger, représenté par la quantité "time". La colonne "status" représente le statut d'arrêt de l'algorithme. Le status first_order signifie notamment que le point atteint est un point vérifiant les conditions KKT du premier ordre (4). En complément du status nous récupérons également la faisabilité primale et duale, voir respectivement les équations (4b) et (4a). Nous pouvons remarquer que sur ce type de problème jouet avec très peu de variables, la méthode du Lagrangien augmenté prend moins de temps. Cependant, nous pouvons constater l'effet du changement de variable $s = Zt$ dans la méthode avec ENVELOPT. En effet, la contrainte est toujours résolue de manière exacte (on rappelle que Z est construit de sorte que $JZ = 0$, ce qui implique $Js = JZt = 0$), ce qui se caractérise par des violations de contraintes nulles (ici on atteint 10^{-16}, qui est l'erreur machine). Ainsi, chaque itération de l'algorithme fournit un point réalisable. Au contraire, la violation des contraintes est plus élevée pour le Lagrangien augmenté car la faisabilité est atteinte au cours des itérations de l'algorithme.

Nous nous intéressons à présent à un problème plus compliqué en utilisant toujours le régulariseur $\psi(x) = \lambda\|x\|_1$. Le problème `clnlbeam` comporte 99 variables et 64 contraintes. Le point initial pour créer le problème quadratique est $x_0 = (1, \ldots, 1)$. Les résultats sont répertoriés dans le tableau 2.

Tableau 2 – Résultats numériques pour un problème quadratique obtenu avec `clnlbeam` et $\psi(x) = \lambda\|x\|_1$.

Méthode	obj	iter	subiter	cviol	time (s)	status	primal_feas	dual_feas
Env-Tron	48.8	10	244	1.37e-14	1.18e-1	first_order	0.0	4.16e-6
Env-Trunk	48.8	10	832	1.43e-14	3.94e-1	first_order	0.0	5.31e-8
Env-IPOPT	48.8	10	63	1.42e-14	7.34e-2	first_order	0.0	1.82e-6
Env-MadNLP	48.8	10	67	1.39e-14	5.99e-2	first_order	0.0	1.84e-6
Lagrangien-R2	1060.0	101	95088	3.44e-7	2.48e-1	max_iter	8.63e-8	76.3
Lagrangien-R2N	1060.0	48	24942	3.91e-8	9.75e1	first_order	3.83e-8	9.51e-6

Sur ce problème de plus grande dimension, nous constatons que le Lagrangien augmenté effectue un nombre beaucoup plus important d'itérations, tout en produisant une violation de contrainte plus élevée que la méthode utilisant ENVELOPT, illustrant l'efficacité de notre méthode pour la résolution de problèmes quadratique.

Nous illustrons à présent les résultats de notre méthode sur des régulariseurs convexes autres que la norme ℓ_1. Le problème `controlinvestment` comporte 100 variables et 50 contraintes. Le point initial pour créer le problème quadratique est $x_0 = (1, \ldots, 1)$. Nous choisissons ici $\psi(x) = \lambda\|x\|_2$ avec $\lambda = 1.5$. Les résultats sont répertoriés dans le tableau 3.

Tableau 3 – Résultats numériques pour un problème quadratique obtenu avec `controlinvestment` et $\psi(x) = \lambda\|x\|_2$.

Méthode	obj	iter	subiter	cviol	time (s)	status	primal_feas	dual_feas
Env-Tron	8.04	8	32	2.08e-15	2.27e-2	first_order	0.0	3.83e-6
Env-Trunk	8.04	3	364	2.09e-15	3.15e-1	first_order	0.0	1.31e-9
Env-IPOPT	8.04	4	4	2.13e-15	1.24e-2	first_order	0.0	3.33e-6
Env-MadNLP	8.04	4	4	2.11e-15	1.17e-2	first_order	0.0	3.52e-6
Lagrangien-R2	7.16	7	4854	0.0466	1.64e-2	error	0.0173	0.000173
Lagrangien-R2N	6.95	6	188	0.0601	3.14e-2	error	0.0225	0.000433

Nous pouvons constater que la méthode ENVELOPT résout efficacement le problème. En particulier la solution ne viole pas les contraintes et le solveur termine en peu d'itérations. Au contraire, le Lagrangien augmenté est mis en défaut : il termine prématurément avec une violation des contraintes élevée. D'autres régulariseurs peuvent être utilisés, comme par exemple $\psi(x) = \left(\sum_i \lambda_i |x_i|^2\right)^{1/2}$ avec $\lambda_i = 0.5$ pour $i = 1, \ldots, 50$ et $\lambda_i = 2.5$ sinon. Nous présentons les résultats pour le problème `hovercraft1d` (100 variables, 50 contraintes).

Tableau 4 – Résultats numériques pour un problème quadratique obtenu avec `hovercraft1d` et $\psi(x) = \left(\sum_i \lambda_i |x_i|^2\right)^{1/2}$.

Méthode	obj	iter	subiter	cviol	time (s)	status	primal_feas	dual_feas
Env-Tron	182.0	8	4860	2.14e-13	1.6	first_order	0.0	7.93e-6
Env-Trunk	182.0	5	9798	2.12e-13	3.05	first_order	0.0	6.38e-9
Env-IPOPT	182.0	8	153	2.09e-13	1.54e-1	first_order	0.0	3.58e-6
Env-MadNLP	182.0	7	120	2.16e-13	1.2e-1	first_order	0.0	3.86e-6
Lagrangien-R2	191.0	101	101000	3.44e-6	2.21e-1	max_iter	5.98e-7	0.374
Lagrangien-R2N	182.0	8	319	2.11e-7	9.4e-1	first_order	1.01e-7	9.94e-6

Nous pouvons constater que les résultats pour ce problème et avec ce nouveau régulariseur sont similaires aux précédents.

4.2.3 Conclusion

La méthode fondée sur ENVELOPT semble converger en un nombre d'itérations inférieur à celui de la méthode du Lagrangien augmenté non lisse. Par ailleurs, les solutions obtenues avec ENVELOPT satisfont systématiquement les contraintes, ce qui n'est pas toujours le cas pour la méthode du Lagrangien augmenté. De plus, l'utilisation de MADNLP ou d'IPOPT comme solveur interne d'ENVELOPT conduit à de meilleures performances que l'utilisation de TRON ou TRUNK. En particulier, ces deux solveurs permettent d'atteindre une solution en un nombre total d'itérations internes plus faible, tout en conservant une bonne faisabilité primale et duale. Enfin, sur les problèmes les plus difficiles (`clnlbeam`, `controlinvestment` et `hovercraft1d`), les temps de calcul obtenus avec IPOPT et MADNLP sont comparables, voire inférieurs, à ceux des autres approches considérées, qu'il s'agisse de la méthode du Lagrangien augmenté non lisse ou des autres variantes d'ENVELOPT. Ces résultats suggèrent qu'ENVELOPT, associé à IPOPT ou MADNLP comme solveur interne, constitue l'approche la plus pertinente pour résoudre la classe de problèmes définie en (18).

4.3 Résolution du problème non lisse sous contraintes

4.3.1 Détails d'implémentation

Nous avons implémenté l'Algorithme 3 à l'aide de packages provenant de l'organisation Julia JuliaSmoothOptimizers.

Résolution du pas normal. Pour résoudre le problème normal (21) (Ligne 10), nous utilisons le solveur LSMR [21] présent dans le package `Krylov.jl` [31] [11] lorsqu'il n'y a pas de contraintes de bornes. Lorsque le problème possède des contraintes de bornes, nous remplaçons la contrainte $\|v_k\|_2 \leq \kappa_v \nu_k \delta_k$ par $\|v_k\|_\infty \leq \kappa_v^\infty \nu_k \delta_k$. Ce changement est motivé par le fait que la contrainte devient simplement une contrainte de borne, ce qui peut être traité très efficacement par des solveurs. En prenant $\kappa_v^\infty \leq \frac{1}{\sqrt{n}}\kappa_v$, alors $\|v_k\|_2 \leq \sqrt{n}\|v_k\|_\infty \leq \sqrt{n}\kappa_v^\infty \nu_k \delta_k \leq \kappa_v \nu_k \delta_k$, ce qui ne change pas les conditions imposées à l'origine. Notons tout de même que ce changement est plus restrictif sur les bornes de v_k. Dans le cas de contraintes de bornes, nous résolvons le problème (21) avec le solveur TronLS présent dans le package `JSOSolver.jl` [29] [12].

Résolution du pas tangent. Pour résoudre le problème tangent (22) (Ligne 12), nous le transformons en le problème (26) que nous résolvons avec Envelopt, comme décrit dans la section 3.2.1.

Estimateur des multiplicateurs de Lagrange. Rappelons les conditions KKT du problème tangent :

$$\begin{aligned} & g_k + g_{h,k} + \frac{u_k}{\nu_k} - J_k^\top y_k = 0 \\ \iff & J_k^\top y_k = g_k + g_{h,k} + \frac{u_k}{\nu_k}. \end{aligned}$$

Nous utilisons un estimateur des moindres carrés pour obtenir les multiplicateurs de Lagrange :

$$y_k = \arg\min \left\| J_k^\top y - \left(g_k + g_{h,k} + \frac{u_k}{\nu_k} \right) \right\|_2^2. \tag{47}$$

Adaptation du paramètre proximal. Dans l'Algorithme 3, le paramètre proximal reste inchangé lorsque le pas est accepté (voir Ligne 19). Dans notre implémentation, nous nous autorisons à le mettre à jour lorsque le pas est très bon pour lui permettre de possiblement prendre des valeurs plus grandes. Pour ce faire nous nous inspirons de la stratégie de mise à jour proposée dans les méthodes de régularisation quadratique dans le cadre non lisse [1, 19]. Nous calculons le ratio

$$\rho_k = \frac{\phi_{\tau_k}(x_k) - \phi_{\tau_k}(x_k + s_k)}{\tilde{\phi}_{\tau_k}(0) - \tilde{\phi}_{\tau_k}(s_k)}$$

où $\tilde{\phi}_{\tau_k}(s)$ est un modèle de la fonction de mérite (23) en x_k tel que

$$\tilde{\phi}_{\tau_k}(s) = \tau(f_k + g_k^\top s + \frac{1}{2\nu_k}\|s\|_2^2 + h(x_k + s_k)) + \|c_k + J_k s\|.$$

Si le ratio ρ_k est plus grand qu'un certain η_ρ, alors nous considérons le pas s_k comme très bon et nous pouvons nous autoriser à réduire la régularisation en actualisant le paramètre proximal : $\nu_{k+1} = \max(1.0, \xi^{-1}\nu_k)$.

Métrique de stationnarité. Lorsque la résolution du problème tangent (22) est exacte, la condition de stationnarité (25) peut se réécrire

$$g_k + g_{h,k} - J_k^\top y_k - z_k = -\frac{1}{\nu_k} s_k. \tag{48}$$

11. JuliaSmoothOptimizers/Krylov.jl
12. JuliaSmoothOptimizers/JSOSolvers.jl

De plus, la mesure de stationnarité du problème (1) est

$$g_k + \tilde{g}_{h,k} - J_k^\top \tilde{y}_k - \tilde{z}_k = 0, \tag{49}$$

où $\tilde{g}_{h,k} \in \partial h(x_k)$, $\tilde{y}_k$ est le multiplicateur associé aux contraintes non linéaires $c(x_k) = 0$ et $\tilde{z}_k$ est le multiplicateur associé aux contraintes de bornes $x_k \in \Omega$. Curtis et al. [14] montrent que $\lim_k \|s_k\| = 0$, et on peut montrer ([14, Lemme 5.1]) que $s_k = 0 \iff u_k = v_k = 0$. Si l'algorithme termine à la Ligne 15 alors $g_k + g_{h,k} - J_k^\top y_k - z_k = 0$ et $g_{h,k} \in \partial h(x_k)$ et z_k correspond aux multiplicateurs associés à $x_k \in \Omega$. Ainsi $g_k + g_{h,k} - J_k^\top y_k - z_k = 0$ représente une condition d'arrêt pour l'Algorithme 3. Cela implique $s_k = 0$, donc $u_k = v_k = 0$. Afin de facilement interpréter notre condition d'arrêt, nous choisissons $u_k = v_k = 0$ comme métrique d'arrêt. En effet, $v_k = 0$ et $u_k \neq 0$ signifie que l'on a atteint la faisabilité, mais que le point n'est pas encore stationnaire. Le cas $v_k \neq 0$ et $u_k = 0$ signifie que l'on a atteint la stationnarité dans l'espace tangent, mais que le point n'est pas encore réalisable. Enfin la situation $v_k = u_k = 0$ représente les conditions KKT optimales.

Nous vérifierons également la faisabilité primale caractérisée par $c(x_k) = 0$. Enfin, nous vérifierons la complémentarité. Si $\Omega = \prod[l_i, u_i]$ alors il s'agit de vérifier pour tout $i \in \{1, \ldots, n\}$ que $\min\{x_i - l_i, z_{k,i}\} = 0$ et $\min\{u_i - x_i, -z_{k,i}\} = 0$. En pratique, nous vérifierons $\|v_k\| \leq \epsilon_v$, $\|u_k\| \leq \epsilon_u$, $\|c(x_k)\| \leq \epsilon_c$ et $\|\min\{x_k - l, z_k\} = 0\| \leq \epsilon_z$, $\|\min\{u - x_k, -z_k\} = 0\| \leq \epsilon_z$.

Vérification des hypothèses. Nous montrons que le sous-solveur ENVELOPT avec la méthode du noyau tel que décrit dans la section 3.1 vérifie les prérequis énoncés en Remarque 5.

1. Les contraintes $J_k u = 0$ sont satisfaites de manière exacte grâce à la méthode du noyau.
2. $g_{h,k} \in \partial h(x_k + v_k + w_k)$ et $\|u_k - w_k\| \leq \epsilon_k$, $\epsilon_k \to 0$ par construction du solveur ENVELOPT.
3. La condition de *décroissance simple* est respectée par ENVELOPT si l'on initialise le problème tangent à zéro en vertu du [17, Lemme 3.2].
4. La condition de *décroissance suffisante* peut être respectée par ENVELOPT. En effet, la condition d'arrêt d'ENVELOPT, donnée en (36), s'écrit

$$\begin{aligned} &\left\| Z_k^\top (g_k + g_{h,k} + \frac{u_k}{\nu_k}) \right\| \leq \epsilon_k. \\ \implies &\left\| Z_k^\top p_k \right\| \leq \epsilon_k \end{aligned}$$

   en notant $p_k = g_k + g_{h,k} + \frac{u_k}{\nu_k}$. Si y_k est un estimateur des moindres carrés comme introduit dans le Lemme 7 alors, $y_k = (J_k J_k^\top)^{-1} J_k p_k$ et donc $r_k = p_k - J_k^\top (J_k J_k^\top)^{-1} J_k p_k$. Si Z_k est une base orthonormale de $\ker(J_k)$ alors d'après le Lemme 4, $(I - Z_k Z_k^\top) p_k = J_k^\top (J_k J_k^\top)^{-1} J_k p_k$ ce qui implique $r_k = Z_k Z_k^\top p_k$. Par conséquent, $\|r_k\| = \|Z_k Z_k^\top p_k\| = \|Z_k p_k\| \leq \epsilon_k$. Enfin, si à l'itération i d'ENVELOPT, $\|r_{k,i}\| \leq \epsilon_{k,i} \leq \frac{\|u_{k,i}\|}{\nu_k}$ alors la condition (33) est respectée, sinon nous effectuons une itération supplémentaire d'ENVELOPT de sorte que $\epsilon_{k,i+1} < \epsilon_{k,i}$. Cela implique $\|r_{k,i}\| \underset{i \to \infty}{\longrightarrow} 0$ et nécessairement $\frac{\|u_{k,i}\|}{\nu_k} > 0$ sinon nous nous serions arrêtés à l'itération précédente de l'Algorithme 3. Un raisonnement similaire permet de vérifier que la condition (32) peut être respectée par ENVELOPT.

Contrôle du résidu. Nous imposons que le pas calculé implique la décroissance suffisante (32). Cependant, il faut également que le résidu tende vers 0 si l'on souhaite avoir un point stationnaire à partir d'une itération. Pour ce faire, lorsque le pas est accepté (voir Ligne 18 de l'Algorithme 3), nous mettons à jour la tolérance minimale avec laquelle ENVELOPT doit résoudre le problème (voir les conditions d'arrêts (36)) : $\epsilon_{Env,k+1} = \xi_{Env} \epsilon_{Env,k}$ où $\xi_{Env} \in (0, 1)$. Au début, la tolérance minimale imposée à ENVELOPT est grande afin de se déplacer rapidement dans une direction de descente sans avoir à résoudre le problème tangent de manière très précise, puis nous réduisons la tolérance au fur et à mesure que les itérations sont acceptées.

Paramètres algorithmiques. Nous répertorions dans le tableau 5 les valeurs des paramètres algorithmiques choisis.

Tableau 5 – Paramètres utilisés dans l'Algorithme 3.

τ_{-1}	κ_v	η_ρ	ξ	ξ_{Env}	η_Φ	σ_c	ϵ_τ	$\epsilon_v = \epsilon_u$	ϵ_c	ϵ_z	κ_r
1.0	10^3	0.7	0.5	0.25	10^{-5}	0.1	0.1	6×10^{-6}	10^{-6}	10^{-4}	0.5

4.3.2 Résultats numériques

Résolution exacte du problème tangent. Dans un premier temps, nous étudions les performances lorsque nous résolvons le problème tangent (22) de manière exacte selon la méthode présentée dans la section 3.2.1. L'objectif est également de vérifier numériquement que notre méthode permet de résoudre des problèmes non lisses avec des régulariseurs différents de la norme ℓ_1. Pour ce faire, nous sélectionnons des problèmes avec contraintes d'égalité parmi le jeu de problèmes mis à disposition dans le package `OptimizationProblems` et nous ajoutons à la fonction objectif un terme non lisse provenant du package `ProximalOperators`. Nous comparons les résultats de notre méthode à ceux d'ENVELOPT utilisé sur le problème initial (1) avec $F(t) = t$ et les paramètres du solveur par défaut.

Nous commençons avec un problème simple avec $h(x) = \|x\|_\infty$ et $\lambda = 10.0$. Le problème `hs56` comporte 7 variables et 4 contraintes. Les résultats sont répertoriés dans le tableau 6.

Tableau 6 – Résultats numériques pour le problème `hs56` et $h(x) = \lambda\|x\|_\infty$.

Méthode	**obj**	**iter**	**itenv**	**subiter**	**time** (s)	**status**	**primal_feas**	**dual_feas**
SQP	3.11e-28	10	121	311	0.052	first_order	2.09e-9	6.68e-6
Envelopt	-7.1e-42		8	48	0.00505	first_order	1.1e-14	1.21e-7

Dans ce tableau, la colonne "Méthode" renseigne la méthode utilisée, *SQP* désignant la méthode proposée dans la section 3.2.1, et *Envelopt* désignant la méthode éponyme avec les paramètres par défaut. Par la suite, lorsque nous ferons référence à cette configuration particulière d'ENVELOPT, nous noterons *Envelopt* la méthode, afin de la distinguer avec le solveur ENVELOPT. Les colonnes "obj", "time", "status", "primal_feas" et "dual_feas" représentent les mêmes informations que dans la section 4.2.2. La colonne "itenv" désigne le nombre d'itérations de l'algorithme ENVELOPT, tandis que "subiter" désigne le nombre d'itération du sous-solveur. La colonne "iter" représente le nombre d'itérations de la boucle générale de l'algorithme SQP. Par conséquent cette information n'existe pas pour la méthode *Envelopt*. Sur cet exemple, nous pouvons constater que notre méthode permet de résoudre le problème non lisse pour un régulariseur différent de la norme ℓ_1, ce qui représente une contribution numérique majeure vis à vis des articles [14] et [15]. En effet, la faisabilité primale et duale est très bonne, attestant que nous avons réussi à atteindre un point stationnaire. De plus, la valeur de la fonction objectif est identique à celle obtenue avec la méthode *Envelopt*. Cependant, le nombre d'itérations nécessaire avant d'arriver à convergence, à la fois pour les itérations d'ENVELOPT et les itérations du sous-solveur, sont plus important pour notre méthode que *Envelopt*. Cela a une répercussion cohérente sur le temps nécessaire à l'algorithme pour converger. Nous discuterons plus en détail de ce point lorsque nous aborderons la résolution inexacte du problème tangent.

Nous illustrons à présent les résultats sur un problème avec plus de variables et de contraintes. Le problème `controlinvestment` comporte 100 variables et 50 contraintes. Nous prenons le régulariseur $h(x) = \left(\sum_i \lambda_i |x_i|^2\right)^{1/2}$ avec $\lambda_i = 5.5$ pour $i = 1, \ldots, 50$ et $\lambda_i = 7.5$ sinon. Les résultats sont répertoriés dans le tableau 7.

Les observations faites sur le problème simple `hs56` restent valables pour ce problème. En particulier, nous sommes capables de résoudre le problème pour un régulariseur autre que la norme ℓ_1 avec *SQP*, la valeur de la fonction objectif est semblable à celle obtenue avec *Envelopt* et le point final permet une faisabilité primale et duale satisfaisante.

Tableau 7 – Résultats numériques pour le problème `controlinvestment` et $h(x) = \left(\sum_i \lambda_i |x_i|^2\right)^{1/2}$.

Méthode	obj	iter	itenv	subiter	time (s)	status	primal_feas	dual_feas
SQP	37.9	3	34	312	0.258	first_order	0.0	4.39e-5
Envelopt	37.9		8	65	0.0191	first_order	9.57e-17	8.83e-10

Plus généralement, nous sélectionnons un ensemble de problèmes provenant d'`Optimization Problems.jl` ayant les caractéristiques suivantes : le problème possède entre 2 et 100 variables, il comporte uniquement des contraintes d'égalités, il comporte moins de contraintes que de variables (c'est-à-dire $m < n$) et la jacobienne des contraintes est de plein rang. Cela permet de constituer un jeu de problèmes de 51 problèmes, que nous résolvons successivement avec notre méthode pour $h(x) = \lambda\|x\|_\infty$. Parmi les 51 problèmes, l'algorithme ne converge pas vers un point stationnaire pour seulement deux d'entre eux (`avion2` et `hovercraft1d`), les autres étant résolus avec succès. Sur les deux problèmes mettant en défaut l'algorithme, ce dernier atteint le nombre maximal d'itérations. Regardons par exemple les résultats obtenus pour `hovercraft1d` avec notre méthode. La valeur finale de la fonction objectif est 1001.83 et la valeur de violation de contraintes est 2.46×10^{-12}. De plus, la faisabilité primale oscille autour de 10^{-6} lors des dernières itérations de l'algorithme. A titre de comparaison, la méthode *Envelopt* donne également 1001.83 comme valeur finale pour la fonction objectif, pour une violation des contraintes de l'ordre de 10^{-14}. Cela laisse penser que notre méthode arrive à se placer dans une région très proche du minimum local mais n'arrive pas à converger de manière à satisfaire les conditions d'arrêt de l'algorithme. Notons respectivement $x^\star_{Envelopt}$ et $x^\star_{SQP}$ les points d'arrêts de la méthode *Envelopt* et de notre méthode. Alors $\|x^\star_{Envelopt} - x^\star_{SQP}\|_\infty / \|x^\star_{Envelopt}\|_\infty = 3.3 \times 10^{-5}$, ce qui représente une erreur relative de 0.003 % en norme infinie par rapport à la solution fournie par *Envelopt*, confirmant la proximité des deux solutions. Cette observation, combinée à la proximité des valeurs de la fonction objectif, renforce l'hypothèse selon laquelle notre méthode parvient à atteindre une région proche d'un minimum local, mais rencontre des difficultés à y converger de manière satisfaisante. Un phénomène similaire a lieu pour le problème `avion2`. Cette observation constitue une piste d'amélioration de l'algorithme : il pourrait être pertinent de détecter les situations dans lesquelles les itérés oscillent sans converger afin d'interrompre prématurément l'algorithme.

Remarque 6. Nous nous appuyons sur la proximité observée entre les points obtenus avec *Envelopt* et notre méthode pour soutenir le raisonnement précédent et conclure que notre algorithme semble avoir atteint une région proche de celle contenant un minimum local. Cette observation suggère ainsi que la non convergence rencontrée par notre méthode est due à une difficulté à satisfaire les conditions d'arrêt dans cette région. Cependant, si les deux points d'arrêt étaient éloignés, une telle conclusion ne pourrait pas être tirée. En effet, la fonction objectif peut admettre plusieurs minima locaux, et *Envelopt* et notre méthode pourraient alors converger vers deux minima locaux distincts. La proximité des deux points constitue donc un élément important pour interpréter les difficultés de convergence observées et le fait que notre algorithme semble tout de même donner une solution satisfaisante.

Résolution inexacte du problème tangent. Nous étudions à présent les performances de notre algorithme lorsque nous résolvons le problème tangent (22) de manière inexacte selon les garanties théoriques de convergence et de complexité établies dans la section 3.2.2. Nous illustrons d'abord numériquement que nous sommes toujours capables de résoudre des problèmes d'optimisation non lisse de la forme (1). Les résultats de convergence sont établis dans le cas où il n'y a pas de contraintes de bornes, c'est-à-dire dans le cas $\Omega = \mathbb{R}^n$. Nous choisissons ainsi 50 problèmes du package `CUTEst.jl` [32], une interface Julia du package Fortran CUTEst [13] mettant à disposition de nombreux problèmes d'optimisation. Les problèmes choisis ne contiennent pas de contraintes de bornes. Nous ajoutons ensuite un terme non lisse en choisissant un opérateur de `ProximalOperators`. Avec $h(x) = \lambda\|x\|_2$ et $\lambda = 5.5$, la méthode proposée permet de trouver un point stationnaire pour 46 problèmes sur 50. Sur les quatre problèmes non résolus, deux (`MSS1` et `VANDANIUMS`) ont été identifiés comme infaisables (voir Ligne 7

13. CUTEst

de l'Algorithme 3), et la méthode *Envelopt* détecte également ces deux problèmes comme infaisables. Le problème `BT4` a été identifié comme non borné, la fonction objectif divergeant vers $-\infty$. La méthode *Envelopt* échoue également avec l'erreur `RESTORATION_FAILED`, qui peut notamment être déclenchée par la non-bornitude du problème. Enfin la méthode proposée n'arrive pas à résoudre le problème `SSINE`, et termine en atteignant le nombre maximal d'itérations sur un point non réalisable (c'est-à-dire $c(x^\star) > \epsilon_c$). La méthode *Envelopt* classe ce problème comme infaisable. Notre algorithme n'est donc pas réellement mis en défaut par le problème. Celui-ci met néanmoins en évidence une piste d'amélioration des conditions d'arrêt.

Comparons à présent les résultats obtenus lorsque l'on résout le problème tangent de manière exacte et de manière inexacte. Nous résolvons le problème `SPIN2` (102 variables et 100 contraintes) avec le régulariseur $h(x) = \lambda\|x\|_1$ et $\lambda = 6.5$. Les résultats sont répertoriés dans le tableau 8, où *SQPi* est la méthode où nous résolvons de manière inexacte le problème tangent (22).

Tableau 8 – Résultats numériques pour le problème `SPIN2` et $h(x) = \lambda\|x\|_1$.

Méthode	obj	iter	itenv	subiter	time (s)	status	primal_feas	dual_feas
SQPi	569.0	30	313	1118	1.1	first_order	2.46e-9	4.4e-6
SQP	569.0	30	250	4042	1.48	first_order	1.5e-9	1.17e-5
Envelopt	569.0	0	13	143	0.194	first_order	3.84e-11	3.96e-7

Notons premièrement que la fonction objectif prend la même valeur pour les trois méthodes, et que la faisabilité primale et duale sont vérifiées. Sur ce problème, nous pouvons constater que les méthodes *SQP* et *SQPi* font le même nombre d'itérations de boucle principale. Le nombre d'itérations d'Envelopt est légèrement plus élevé pour la méthode inexacte. Cependant, malgré un nombre plus élevé d'itérations d'Envelopt, la méthode *SQPi* fait presque quatre fois moins d'itérations du sous-solveur d'Envelopt, illustrant le bénéfice de la résolution inexacte du problème tangent (nous rappelons que c'est ce dernier qui est résolu avec Envelopt). En effet, en permettant de relâcher la précision de résolution d'Envelopt, ce dernier a besoin de moins d'itérations pour atteindre la tolérance minimale imposée. Cela résulte logiquement en un nombre d'itérations du sous-solveur plus faible. Ce nombre est toutefois encore plus élevé que la méthode *Envelopt*.

Afin d'illustrer ces observations entre les méthodes *SQP* et *SQPi* sur l'ensemble des problèmes testés, nous utilisons les profils de performances introduits par Dolan et Moré [20]. Cette approche permet de synthétiser les résultats d'un ensemble de solveurs au moyen d'une représentation graphique reposant sur deux quantités. Pour une métrique de performance donnée (le temps de calcul, le nombre d'itérations ...) que nous notons *cost*, le ratio de performance associé au solveur s pour un problème p est défini par

$$\tau(s,p) = \frac{\text{cost}(s,p)}{\min_{s'} \text{cost}(s',p)} \tag{50}$$

où le dénominateur correspond à la meilleure performance obtenue par l'un des solveurs sur le problème considéré. Par conséquent, $\tau(s,p) = 1$ signifie que le solveur s obtient la meilleure performance pour le problème p, tandis que $\tau(s,p) = 2$ signifie que son coût est deux fois supérieur à celui du meilleur solveur. Cette quantité est l'abscisse du profil de performance. L'ordonnée du profil, notée $\rho_s(\tau)$, représente la fraction des problèmes pour lesquels le solveur s atteint un ratio de performance inférieur ou égal à τ :

$$\rho_s(\tau) = \frac{\#\{p : \tau(s,p) \leq \tau\}}{\#\{\text{problèmes considérés}\}}. \tag{51}$$

De cette manière $\rho_s(1)$ correspond à la proportion de problèmes pour lesquels le solveur s est le plus performant. Lorsque τ augmente, $\rho_s(\tau)$ indique la proportion de problèmes pour lesquels le solveur reste compétitif par rapport au meilleur solveur. La valeur finale asymptotique à droite donne le taux de succès global du solveur (c'est le ratio entre le nombre de problèmes résolus par le solveur et le nombre total de problèmes). Si la courbe atteint 1, alors le solveur résout tous les problèmes, mais

sans aucune garanti de rapidité. Un solveur avec une courbe en haut à gauche est préférable par cela indique que ce solveur est le plus efficace sur beaucoup de problèmes que les autres solveurs.

Afin de tracer les profils de performance, nous utilisons le package `SolverBenchmark.jl` [36]. Les profils de performance sont construits pour les méthodes *SQP* (en bleu) et *SQPi* (en orange) sur les 50 problèmes CUTEst mentionnés précédemment. Nous présentons ces profils pour $h(x) = \lambda\|x\|_1$ sur la figure 1, $h(x) = \lambda\|x\|_\infty$ sur la figure 2 et $h(x) = \lambda\|x\|_2$ sur la figure 3.

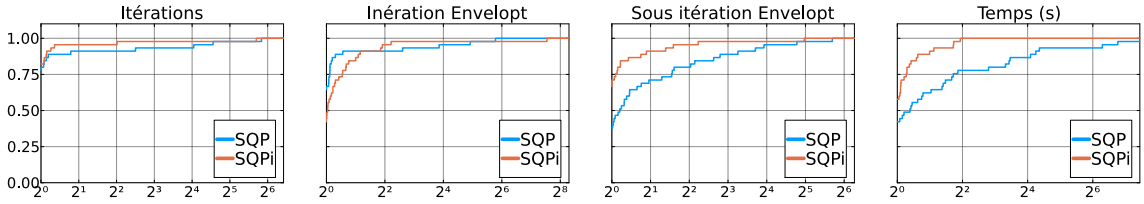


Figure 1 – Profil de performance sur 50 problèmes CUTEst pour $h(x) = \lambda\|x\|_1$, $\lambda = 6.5$.
***En bleu :* méthode SQP. *En orange :* méthode SQP avec résolution inexacte du problème tangent (22).**

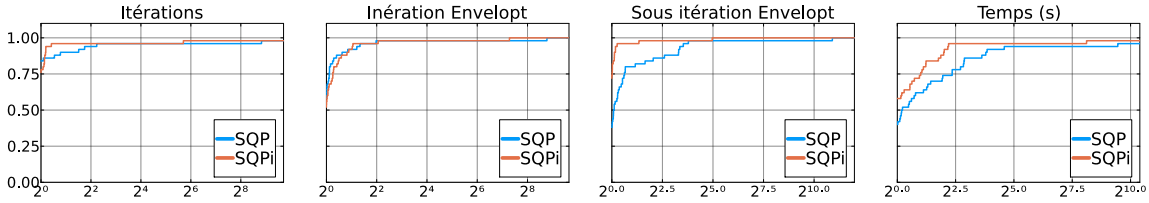


Figure 2 – Profil de performance sur 50 problèmes CUTEst pour $h(x) = \lambda\|x\|_\infty$, $\lambda = 10.0$.
***En bleu :* méthode SQP. *En orange :* méthode SQP avec résolution inexacte du problème tangent (22).**

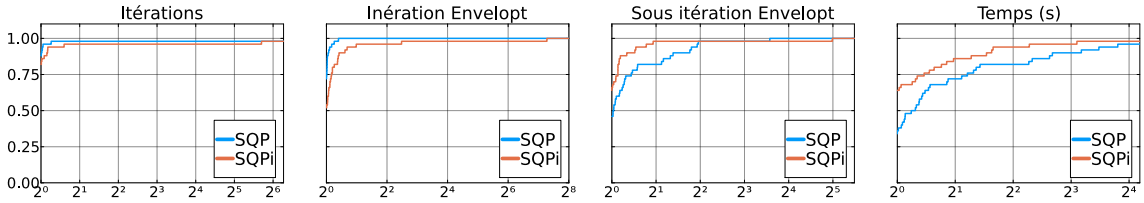


Figure 3 – Profil de performance sur 50 problèmes CUTEst pour $h(x) = \lambda\|x\|_2$, $\lambda = 5.5$.
***En bleu :* méthode SQP. *En orange :* méthode SQP avec résolution inexacte du problème tangent (22).**

Nous nous intéressons au nombre total d'itérations de l'Algorithme 3 (premier graphe), au nombre total d'itérations d'ENVELOPT pour résoudre (22) (deuxième graphe), au nombre d'itérations du sous-solveur d'ENVELOPT pour résoudre ce même sous-problème (troisième graphe), ainsi qu'au temps d'exécution nécessaire à la résolution du problème (quatrième graphe). Pour les trois régulariseurs, les deux méthodes présentent des performances similaires en termes de nombre d'itérations global et de nombre d'itérations d'ENVELOPT. Cependant, la méthode *SQPi* s'avère plus performante en termes de nombre d'itérations du sous-solveur d'ENVELOPT. En effet, la partie gauche du profil de la méthode *SQPi* (en orange) est nettement supérieure à celle de la méthode *SQP* (en bleu), indiquant que le sous-solveur nécessite significativement moins d'itérations avec la méthode *SQPi*. Cette réduction du nombre d'itérations du sous-solveur a un impact direct sur le temps de résolution des problèmes : moins d'itération du sous-solveur indique une convergence plus rapide. Cette amélioration est visible sur le profil de la méthode *SQPi* (en orange), situé au-dessus de celui de la méthode *SQP* (en bleu). L'observation faite sur le problème `SPIN2` (voir table 8) se confirme globalement sur les 50 problèmes étudiés. La résolution de problème tangent (22) permet ainsi de diminuer significativement le nombre d'itérations du sous-solveur d'ENVELOPT sur la majorité des problèmes, permettant une réduction du temps de résolution.

5 Conclusions et perspectives

La résolution d'un problème non lisse avec contraintes non linéaires représente un défi complexe. Lorsqu'il n'y a pas de terme non lisse, le problème peut notamment être résolu à l'aide d'une méthode du Lagrangien augmenté ou de la méthode Sequential Quadratic Programming. Lorsqu'il n'y a pas de contraintes, il est par exemple possible d'utiliser des méthodes de régularisation quadratique pour l'optimisation non lisse. Curtis et al. [14] proposent une variante de la méthode SQP pour traiter les problèmes non lisses avec contraintes d'égalité. Cependant, les tests numériques effectués dans cet article se limitent à la norme ℓ_1 comme terme non lisse, car celle-ci peut être reformulée sous la forme d'un problème lisse. Nous avons donc proposé une amélioration de leur méthode afin de pouvoir traiter efficacement, sans transformation préalable, d'autres régulariseurs non lisses. La modification consiste essentiellement à utiliser le solveur ENVELOPT couplé, à une méthode de noyau, pour résoudre le sous-problème tangent de la méthode SQP. Nous avons également montré, en l'absence de contraintes de bornes, que les résultats de convergence de l'algorithme sont préservés lorsque le sous-problème tangent est résolu de manière inexacte avec notre nouvelle méthode. Enfin, nous avons montré que l'hypothèse de bornitude uniforme du sous-différentiel de h, telle qu'imposée dans [14] et [15], peut être relâchée en ne requérant qu'une bornitude du sous-gradient de norme minimale le long des itérés, élargissant ainsi la classe de fonctions non lisses admissibles au-delà des seules fonctions à sous-différentiel borné. Ces résultats théoriques ont été validés expérimentalement. D'une part, notre méthode résout efficacement des problèmes quadratiques similaires au sous-problème tangent rencontré dans la méthode générale. D'autre part, elle permet de traiter des problèmes non lisses faisant intervenir une variété de régulariseurs convexes, au-delà de la seule norme ℓ_1. Enfin, nous avons montré l'intérêt de la résolution inexacte du sous-problème tangent. La méthode proposée présente cependant certaines limites, ouvrant plusieurs pistes d'amélioration, notamment concernant les conditions d'arrêt de l'algorithme dans certains cas particuliers. De plus, les hypothèses formulées dans [14] sont des hypothèses fortes (matrice J_k de rang plein, fonction h convexe, ...), et un travail supplémentaire pourrait permettre de les alléger. Il est également possible d'étudier la résolution inexacte du sous-problème normal ou l'ajout de l'information du second ordre dans le sous-problème tangent. Enfin, une piste d'amélioration majeure consisterait à étendre la méthode aux contraintes d'inégalités pour notamment traiter les contraintes de bornes. Pour ce faire, il serait possible de s'inspirer des méthodes de points intérieurs telles qu'IPOPT.

A Notations

Le tableau A1 résume les notations et symboles principalement utilisés dans le document

Tableau A1 – Notations et symboles.

Terme	Description
x_k	Itéré à l'itération k
$x_\star$	Solution de l'algorithme
f	Partie lisse de la fonction objectif
h	Partie non lisse de la fonction objectif
φ	Modèle de la fonction f
ψ	Modèle de la fonction h
f_k	Fonction f évalué en x_k
c_k	Contraintes c évalué en x_k
g_k	Gradient de la fonction f évalué en x_k
$J(x)$	Jacobienne des contraintes évaluée en x
J_k	Jacobienne des contraintes évaluée en x_k
$H(x)$	Hessienne de la fonction f évaluée en x
H_k	Hessienne de la fonction f évaluée en x_k
Z_k	Base du noyau de J_k
y	Multiplicateur de Lagrange associés aux contraintes d'égalités
z	Multiplicateur de Lagrange associés aux contraintes de borne
$\mathcal{L}(x, y)$	Lagrangien d'un problème avec contraintes
$\mathcal{L}(x, y, \mu)$	Lagrangien augmenté d'un problème avec contraintes
v_k	Solution du k-ième problème normal SQP
u_k	Solution du k-ième problème tangent SQP
s_k	Pas candidat à la k-ième itération de l'algorithme
τ_k	Paramètre de la fonction de mérite
ν_k	Paramètre de régularisation
ϕ	Fonction de mérite
$\overline{\phi}$	Fonction de mérite shiftée
ϵ_v	Tolérance sur la norme du pas normal
ϵ_u	Tolérance sur la norme du pas tangent
ϵ_c	Tolérance sur la violation des contraintes d'égalités
ϵ_z	Tolérance sur la violation des contraintes de bornes
$\epsilon_{Env,k}$	Tolérance du solveur Envelopt à l'itération k
Ω	Contraintes de bornes

B Reformulation de problèmes non lisses en problèmes lisses sous contraintes

Il est possible de transformer un problème d'optimisation faisant intervenir un terme non lisse en un problème d'optimisation lisse avec contraintes. Nous détaillons ici la procédure pour $h(x) = \lambda\|x\|_1$ et $h(x) = \lambda\|x\|_\infty$. Pour alléger les notations, nous ne considérons que des problèmes sans contraintes, mais le raisonnement reste strictement identique lorsque le problème contient des contraintes.

Norme ℓ_1 : Si l'on souhaite résoudre le problème d'optimisation $\min_x \; f(x) + \lambda\|x\|_1$, alors il est équivalent de résoudre le problème

$$\min_{x,a,b} f(x) + \lambda \sum_i (a_i + b_i), \quad \text{tel que} \quad \begin{cases} a, b \geq 0 \\ x - (a + b) = 0 \end{cases}$$

en remarquant que $\|x\|_1 = \sum|x_i|$ et $|x| = \max\{x, 0\} - \min\{x, 0\}$. Le problème reformulé est alors un problème lisse avec contraintes.

Norme ℓ_∞ : Si l'on souhaite résoudre le problème d'optimisation $\min_x \; f(x) + \lambda\|x\|_\infty$, alors il est équivalent de résoudre le problème

$$\min_{x,t} f(x) + \lambda t, \quad \text{tel que} \quad -te \leq x \leq te$$

où $e = (1, \dots, 1)$. Le problème reformulé est alors un problème lisse avec contraintes.

C Algorithmes

C.1 Algorithme du Lagrangien augmenté

Nous présentons ici l'algorithme de Lagrangien augmenté introduit dans la section 2.2.2.

Algorithme C1 Algorithme de Lagrangien Augmenté.

Entrée : Paramètres $\mu_0 > 0$, tolérance $\tau_0 > 0$, points initiaux x_0^s et y_0.

pour $k = 0, 1, 2, \dots$ **faire :**
  Trouver un minimiseur approximatif x_k de $\mathcal{L}_A(\cdot, y_k; \mu_k)$, en partant de x_k^s, tel que

$$\|\nabla_x \mathcal{L}(x_k, y_k; \mu_k)\| \leq \tau_k.$$

  si un critère de convergence pour (2) est satisfait **alors**
    Arrêter avec la solution approximative x_k.
  fin si
  Mettre à jour les multiplicateurs de Lagrange selon(6) afin d'obtenir y_{k+1}.
  Choisir un nouveau paramètre de pénalisation $\mu_{k+1} \geq \mu_k$.
  Définir le point de départ de l'itération suivante :

$$x_{k+1}^s = x_k.$$

  Sélectionner la nouvelle tolérance τ_{k+1}.
fin pour

C.2 Algorithme de pénalisation quadratique non lisse

Nous présentons ici les algorithmes R2 et R2N, respectivement énoncés dans [1] et [19]. Ces algorithmes permettent de résoudre des problèmes non lisses sans contraintes.

Algorithme C2 R2 : Algorithme de régularisation quadratique non lisse.

Entrée : Paramètres $0 < \eta_1 \leq \eta_2 < 1$ et $0 < \gamma_3 \leq 1 < \gamma_1 \leq \gamma_2$.

Choisir $x_0 \in \mathbb{R}^n$ tel que $h(x_0)$ soit fini, $\sigma_0 > 0$, et calculer $f(x_0) + h(x_0)$.
pour $k = 0, 1, 2, \dots$ **faire :**
  Définir $m(s; x_k, \sigma_k)$.
  Calculer un pas s_k selon (15).
  Calculer le ratio

$$\rho_k := \frac{f(x_k) + h(x_k) - \big(f(x_k + s_k) + h(x_k + s_k)\big)}{\varphi(0; x_k) + \psi(0; x_k) - \big(\varphi(s_k; x_k) + \psi(s_k; x_k)\big)}.$$

  si $\rho_k \geq \eta_1$ **alors**
    $x_{k+1} = x_k + s_k$
  sinon
    $x_{k+1} = x_k$
  fin si
  Mettre à jour le paramètre de régularisation :

$$\sigma_{k+1} \in \begin{cases} [\gamma_3 \sigma_k, \sigma_k] & \text{si } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_1 \sigma_k] & \text{si } \eta_1 \leq \rho_k < \eta_2, \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{si } \rho_k < \eta_1. \end{cases}$$

fin pour

Algorithme C3 R2N : Méthode de régularisation quadratique non lisse modifiée.

Entrée : Paramètres $0 < \eta_1 \leq \eta_2 < 1$, $0 < \gamma_3 \leq 1 < \gamma_1 \leq \gamma_2$ et $0 < \theta_1 < 1 < \theta_2$

Choisir $x_0 \in \mathbb{R}^n$ tel que $h(x_0)$ soit fini, $\sigma_0 > 0$, et calculer $f(x_0) + h(x_0)$
pour $k = 0, 1, 2, \ldots$ **faire :**
  Choisir $B_k = B(x_k) \in \mathbb{R}^{n \times n}$ tel que $B_k = B_k^\top$.
  Calculer $\mu_k = \theta_1 / (\|B_k\| + \sigma_k)$.
  Calculer $s_{k,\text{cp}} \in \operatorname{argmin}_s m_{\text{cp}}(s; x_k, \nu_k^{-1})$ et $\xi_{\text{cp}}(s_{k,\text{cp}}; x_k, \nu_k^{-1})$.
  Calculer un pas s_k tel que $m(s_k; x_k, \sigma_k) \leq m(s_{k,\text{cp}}; x_k, \sigma_k)$.
  si $\|s_k\| > \theta_2 \|s_{k,\text{cp}}\|$ **alors**
    Réinitialiser $s_k = s_{k,\text{cp}}$
  fin si
  Calculer le ratio

$$\rho_k := \frac{f(x_k) + h(x_k) - \big(f(x_k + s_k) + h(x_k + s_k)\big)}{\varphi(0; x_k) + \psi(0; x_k) - \big(\varphi(s_k; x_k) + \psi(s_k; x_k)\big)}.$$

  si $\rho_k \geq \eta_1$ **alors**
    $x_{k+1} = x_k + s_k$
  sinon
    $x_{k+1} = x_k$
  fin si
  Mettre à jour le paramètre de régularisation :

$$\sigma_{k+1} \in \begin{cases} [\gamma_3 \sigma_k, \sigma_k] & \text{si } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_1 \sigma_k] & \text{si } \eta_1 \leq \rho_k < \eta_2, \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{si } \rho_k < \eta_1. \end{cases}$$

fin pour

D Lagrangien augmenté et pénalité quadratique non lisse

Nous décrivons ici une méthode de Lagrangien augmenté pour la résolution d'un problème d'optimisation non lisse avec contraintes de la forme (1). A chaque itération, on résoud le sous-problème

$$\min_x \ \mathcal{L}(x, y, \mu) + h(x) \tag{D1}$$

où y est un estimé des multiplicateurs de Lagrange pour les contraintes, μ est le paramètre de pénalisation et $\mathcal{L}(\cdot, y, \mu)$ est le Lagrangien augmenté défini par (5). Afin de résoudre le problème non lisse sans contraintes (D1), nous utilisons l'algorithme R2 [1] ou R2N [19] (voir Annexe C.2). Nous utilisons l'implémentation Julia des algorithmes présents dans le package `RegularizedOptimization` [22] [14].

E Paramètre de la fonction de mérite

A chaque itération, le paramètre de la fonction de mérite τ est ajusté de sorte à ce que s_k soit une direction de descente pour la fonction de mérite et vérifiant $\tau_k \leq \tau_{k-1}$ [14, 15]. Un candidat possible est, pour $\sigma_c \in (0, 1)$,

$$\tau_{k,\text{trial}} = \begin{cases} \infty & \text{si } g_k^\top s_k + \frac{1}{2\nu_k}\|s_k\|_2^2 + h(x_k + s_k) - h(x_k) \leq 0 \\ \frac{(1-\sigma_c)(\|c_k\|_2 - \|c_k + J_k v_k\|_2)}{g_k^\top s_k + \frac{1}{2\nu_k}\|s_k\|_2^2 + h(x_k + s_k) - h(x_k)} & \text{sinon.} \end{cases} \tag{E2}$$

En choisissant $\epsilon_\tau \in (0, 1)$, nous ajustons le paramètre de mérite avec la mise à jour

$$\tau_k = \begin{cases} \tau_{k-1} & \text{si } \tau_{k-1} \leq \tau_{k,\text{trial}} \\ \min\{(1 - \epsilon_\tau)\tau_{k-1}, \tau_{k,\text{trial}}\} & \text{sinon.} \end{cases} \tag{E3}$$

La mise à jour (E3) assure que si le paramètre de mérite décroît à l'itération k, alors il décroit d'au moins une fraction de la valeur précédente.

14. JuliaSmoothOptimizers/RegularizedOptimization.jl

Algorithme D4 Méthode du Lagrangien augmenté non lisse.

Entrée : $\varepsilon, \varepsilon_c > 0$, $\rho > 1$, $\alpha, \beta \in (0,1)$.

Choisir un point initial x_0, des multiplicateurs initiaux y_0, un paramètre de pénalisation initial $\mu_0 > 0$ et une tolérance initiale $\tau_0 > 0$
Calculer $f(x_0)$, $c(x_0)$ et $h(x_0)$; fixer $\theta_{\text{old}} = +\infty$
pour $k = 0, 1, 2, \ldots$ **faire :**
  Fixer la tolérance du sous-problème à $\tau_k = \max(\tau_k, \varepsilon)$
  Résoudre approximativement, avec une tolérance τ_k et en partant de x_k, le problème suivant à l'aide du sous-solveur R2 ou R2N (voir section 2.4) :

$$x_{k+1} \approx \arg\min_x \; \mathcal{L}(x, y_k; \mu_k) + h(x)$$

  Calculer $f(x_{k+1})$, $c(x_{k+1})$ et $h(x_{k+1})$
  Mettre à jour les multiplicateurs selon (6)
  Calculer la violation de faisabilité primale $\theta_{k+1} = \|c(x_{k+1})\|_\infty$
  si le sous-problème a atteint la stationnarité du premier ordre **et** $\tau_k \leq \varepsilon$ **et** $\theta_{k+1} \leq \varepsilon_c$ **alors**
    Sortie : (x_{k+1}, y_{k+1})
  fin si
  si le sous-problème a atteint la stationnarité du premier ordre **alors**
    $\tau_{k+1} = \alpha\, \tau_k$
  sinon
    $\tau_{k+1} = \tau_k$
  fin si
  si $\theta_{k+1} > \max\big(\varepsilon_c,\ \beta\, \theta_{\text{old}}\big)$ **alors**
    $\mu_{k+1} = \rho\, \mu_k$
  sinon
    $\mu_{k+1} = \mu_k$
  fin si
  $\theta_{\text{old}} = \theta_{k+1}, \quad k = k+1$
fin pour
Sortie : (x_{k+1}, y_{k+1})

F Résultats

Cette section regroupe les résultats démontrés dans l'article de Dai et al. [15] et qui sont utilisés dans la section 3.2.2.

Lemme 9. ([15, Lemme 3.4]) Le choix de τ_k dans (E3) garantit que la direction s_k vérifie

$$D_{\Phi_{\tau_k}}(x_k, s_k) \leq -\frac{\tau_k}{2\nu_k}\|s_k\|_2^2 - \sigma_c\big(\|c_k\|_2 - \|c_k + J_k v_k\|_2\big) < 0,$$
$$\Delta q_k(s_k, \tau_k) \geq \frac{\tau_k}{2\nu_k}\|s_k\|_2^2 + \sigma_c\big(\|c_k\|_2 - \|c_k + J_k v_k\|_2\big) > 0.$$

Lemme 10. ([15, Lemme 3.6]) Le pas $s_k = v_k + u_k$ vérifie

$$\|c_k\|_2 - \|c_k + J_k s_k\|_2 = \|c_k\|_2 - \|c_k + J_k v_k\|_2 \geq \frac{1}{2\kappa_c}\|J_k^\top c_k\|_2^2 \min\left\{\frac{1}{1+\kappa_{\nabla c}^2}, \kappa_v \nu_k\right\}.$$

Lemme 11. ([15, Lemme 3.8]) Pour tout $k \geq 1$, on a $0 < \tau_k \leq \tau_{k-1}$.

Lemme 12. ([15, Lemme 3.12]) Pour tout $k \in \mathbb{N}$, on a $\tau_{k,\text{trial}} \geq \tau_{\min,\text{trial}}$, où

$$\tau_{\min,\text{trial}} := \min\left\{\frac{(1-\sigma_c)\kappa_v \sigma_{\min}^2}{2\kappa_v\kappa_{\nabla c}\big(\kappa_{\nabla f} + \kappa_{\partial h} + 1/2 \times \kappa_c\kappa_v\kappa_{\nabla c}\big)}, \frac{(1-\sigma_c)(\sigma_{\min}/\kappa_{\nabla c})^2}{2\kappa_v\kappa_{\nabla c}\big(\kappa_{\nabla f} + \kappa_{\partial h} + 1/2 \times \kappa_c\kappa_v\kappa_{\nabla c}\big)\nu_0}\right\}.$$

En combinant ce résultat avec $v_k^c = \min\left\{\frac{\|J_k^\top c_k\|_2^2}{\|J_k J_k^\top c_k\|_2^2}, \kappa_v\nu_k\right\} J_k^\top c_k$, on obtient

$$\tau_k \geq \tau_{\min} := \min\left\{\tau_0, (1-\epsilon_\tau)\tau_{\min,\text{trial}}\right\}.$$

Lemme 13. ([15, Lemme 3.13]) Si

$$\nu_k \leq \frac{\tau_{\min}}{\tau_{\min} L_g + L_J},$$

alors $k \in \mathcal{S}$. Par conséquent,

$$\nu_k \geq \nu_{\min} := \min\left\{\nu_0, \frac{\xi\tau_{\min}}{\tau_{\min}L_g + L_J}\right\} > 0 \qquad \text{pour tout } k \in \mathbb{N},$$

et le nombre d'itérations infructueuses est borné par

$$|\{k \in \mathbb{N} : k \notin \mathcal{S}\}| \leq \max\left\{0, \left\lceil \frac{\log\left(\frac{\tau_{\min}}{\nu_0(\tau_{\min}L_g + L_J)}\right)}{\log(\xi)} \right\rceil\right\}.$$

Lemme 14. ([15, Lemme 3.14]) Les propriétés suivantes sont vérifiées par la fonction de mérite shiftée

$$\bar{\phi}_\tau(x) := \tau\big(f(x) - f_{\inf} + r(x)\big) + \|c(x)\|_2 :$$

(i) Pour tous $\{x, y\} \subset \mathbb{R}^n$ et $\tau \in \mathbb{R}_{>0}$, on a

$$\bar{\phi}_\tau(x) - \bar{\phi}_\tau(y) = \phi_\tau(x) - \phi_\tau(y).$$

(ii) Pour tout $x \in \mathbb{R}^n$ et $0 < \tau_2 \leq \tau_1$, on a

$$\bar{\phi}_{\tau_2}(x) \leq \bar{\phi}_{\tau_1}(x).$$

(iii) La suite $\{\bar{\phi}_{\tau_k}(x_k)\}$ est monotone décroissante.

Références

[1] Aleksandr Y. Aravkin, Robert Baraldi et Dominique Orban : A Proximal Quasi-Newton Trust-Region Method for Nonsmooth Regularized Optimization. SIAM Journal on Optimization, 32(2):900–929, 2022.

[2] Heinz H. Bauschke et Patrick L. Combettes : Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, 2011.

[3] Dimitri P. Bertsekas : Constrained Optimization and Lagrange Multiplier Methods. Athena Scientific, 1996.

[4] Jeff Bezanson, Alan Edelman, Stefan Karpinski et Viral B. Shah : Julia : A fresh approach to numerical computing. SIAM Review, 59(1):65–98, 2017.

[5] Paul T. Boggs et Jon W. Tolle : Sequential quadratic programming. Acta Numerica, 4:1–51, 1995.

[6] Alfredo Buttari et Alexis Montoison : QRMumps.jl : A Julia interface to qr_mumps, avril 2021. URL https://github.com/JuliaSmoothOptimizers/QRMumps.jl.

[7] Richard H. Byrd : Robust trust region methods for constrained optimization. *In* Third SIAM Conference on Optimization, Houston, Texas, 1987.

[8] Richard H. Byrd, Frank E. Curtis et Jorge Nocedal : An Inexact SQP Method for Equality Constrained Optimization. SIAM Journal on Optimization, 19(1):351–369, 2008.

[9] Richard H. Byrd, Jorge Nocedal et Richard A. Waltz : KNITRO : An integrated package for nonlinear optimization. *In* Gianni di Pillo et Massimo Roma, éditeurs : Large-Scale Nonlinear Optimization, pages 35–59. Springer-Verlag, 2006.

[10] Jian-Feng Cai, Emmanuel J. Candès et Zuowei Shen : A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.

[11] Emilie Chouzenoux, Marie-Caroline Corbineau et Jean-Christophe Pesquet : A proximal interior point algorithm with applications to image processing. Journal of Mathematical Imaging and Vision, 62 (6):919–940, 2020.

[12] Andrew R. Conn, Nicholas I. M. Gould et Philippe L. Toint : Trust Region Methods. Society for Industrial and Applied Mathematics, 2000.

[13] Richard Courant : Variational methods for the solution of problems of equilibrium and vibrations. Bulletin of the American Mathematical Society, 49(1):1–23, 1943.

[14] Frank E. Curtis, Xiaoyi Qu et Daniel P. Robinson : A Proximal-Gradient Method for Solving Regularized Optimization Problems with General Constraints, 2026.

[15] Yutong Dai, Xiaoyi Qu et Daniel P. Robinson : A Proximal-Gradient Method for Equality Constrained Optimization. SIAM Journal on Optimization, 35(4):2654–2683, 2025.

[16] Alberto De Marchi, Xiaoxi Jia, Christian Kanzow et Patrick Mehlitz : Constrained composite optimization and augmented Lagrangian methods. arXiv preprint arXiv :2203.05276, 2022.

[17] Alberto De Marchi et Patrick Mehlitz : Local properties and augmented Lagrangians in fully nonconvex composite optimization. Journal of Nonsmooth Analysis and Optimization, 5, mai 2024.

[18] Alberto De Marchi et Dominique Orban : Envelopt : Constrained convex composite optimization. Cahier du GERAD G-2026-42, GERAD, Montréal, Canada, 2026.

[19] Youssef Diouane, Mohamed L. Habiboullah et Dominique Orban : A proximal modified quasi-newton method for nonsmooth regularized optimization. SIAM Journal on Optimization, 36(2):534–563, 2026. URL https://doi.org/10.1137/24M169761X.

[20] Elizabeth D. Dolan et Jorge J. Moré : Benchmarking optimization software with performance profiles. Mathematical Programming, 91(2):201–213, 2002.

[21] David Chin-Lung Fong et Michael A. Saunders : LSMR : An iterative algorithm for sparse least-squares problems. SIAM Journal on Scientific Computing, 33(5):2950–2971, 2011.

[22] Mathieu Gollier, Mohamed Laghdaf Habiboullah, Geoffroy Leconte, Robert Baraldi, Dominique Orban et Youssef Diouane : RegularizedOptimization.jl : Algorithms for regularized optimization. Journal of Open Source Software, 11(118):9344, 2026.

[23] Nicholas I. M. Gould : On the accurate determination of search directions for simple differentiable penalty functions. IMA Journal of Numerical Analysis, 6:357–372, 1986.

[24] Magnus R. Hestenes : Multiplier and gradient methods. Journal of Optimization Theory and Applications, 4:303–320, 1969.

[25] Marucha Lalee, Jorge Nocedal et Todd Plantenga : On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization. SIAM Journal on Optimization, 8(3):682–706, 1998.

[26] Geoffroy Leconte, Dominique Orban et Abel Soares Siqueira : QuadraticModels.jl : Data structures for linear and quadratic optimization problems based on NLPModels, 2023. URL https://github.com/JuliaSmoothOptimizers/QuadraticModels.jl.

[27] Jinchao Li, Martin S. Andersen et Lieven Vandenberghe : Inexact proximal Newton methods for self-concordant functions. Mathematical Methods of Operations Research, 85(1):19–41, 2017.

[28] Chih-Jen Lin et Jorge J. Moré : Newton's method for large bound-constrained optimization problems. SIAM Journal on Optimization, 9(4):1100–1127, 1999.

[29] Tangi Migot, Dominique Monnet, Dominique Orban et Abel Soares Siqueira : JSOSolvers.jl : Unconstrained and bound-constrained optimization solvers. Journal of Open Source Software, 11(117):9467, 2026.

[30] Tangi Migot, Dominique Orban et Abel Soares Siqueira : OptimizationProblems.jl : A collection of optimization problems in Julia, 2023. URL https://github.com/JuliaSmoothOptimizers/OptimizationProblems.jl.

[31] Alexis Montoison et Dominique Orban : Krylov.jl : A Julia basket of hand-picked Krylov methods. Journal of Open Source Software, 8(89):5187, 2023.

[32] Alexis Montoison, Dominique Orban et Abel Soares Siqueira : CUTEst.jl : Julia's CUTEst interface, 2024. URL https://github.com/JuliaSmoothOptimizers/CUTEst.jl.

[33] Jorge Nocedal et Stephen J. Wright : Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2 édition, 2006.

[34] Emmanuel O. Omojokun : Trust region algorithms for optimization with nonlinear equality and inequality constraints. Thèse de doctorat, University of Colorado, Boulder, 1989.

[35] Dominique Orban et Abel Soares Siqueira : NLPModelsIpopt.jl : A thin IPOPT wrapper for NLPModels, 2023. URL https://github.com/JuliaSmoothOptimizers/NLPModelsIpopt.jl.

[36] Dominique Orban et Abel Soares Siqueira : SolverBenchmark.jl : Benchmark tools for solvers, 2023. URL https://github.com/JuliaSmoothOptimizers/SolverBenchmark.jl.

[37] Neal Parikh et Stephen Boyd : Proximal algorithms. Foundations and Trends in Optimization, 1:127–239, 2014.

[38] R. Tyrrell Rockafellar : Convex Analysis. Princeton University Press, 1970. ISBN 9780691015866.

[39] R. Tyrrell Rockafellar et Roger J.-B. Wets : Variational Analysis. Springer, 1998.

[40] Sungho Shin, Mihai Anitescu et François Pacaud : Accelerating optimal power flow with GPUs : SIMD abstraction of nonlinear programs and condensed-space interior-point methods. Electric Power Systems Research, 236:110651, 2024.

[41] Sungho Shin, Carleton Coffrin, Kaarthik Sundar et Victor M. Zavala : Graph-based modeling and decomposition of energy infrastructures. IFAC-PapersOnLine, 54(3):693–698, 2021.

[42] Gilbert Strang : Linear Algebra and Its Applications. Brooks/Cole, Cengage Learning, 4 édition, 2006. ISBN 9780030105678.

[43] Andreas Wächter et Lorenz T. Biegler : On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 106(1):25–57, 2006.

[44] Robert B Wilson : A simplicial algorithm for concave programming. Ph. D. Dissertation, Graduate School of Bussiness Administration, 1963.

[45] Yuqian Zhang, Yenson Lau, Han-wen Kuo, Sky Cheung, Abhay Pasupathy et John Wright : On the global geometry of sphere-constrained sparse blind deconvolution. *In* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4894–4902, 2017.

[46] Hui Zou et Lingzhou Xue : A selective overview of sparse principal component analysis. Proceedings of the IEEE, 106(8):1311–1320, 2018.