



Titre: Méthodes de région de confiance pour l'optimisation non lisse
Title:

Auteur: Geoffroy Leconte
Author:

Date: 2024

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Leconte, G. (2024). Méthodes de région de confiance pour l'optimisation non lisse
Citation: [Thèse de doctorat, Polytechnique Montréal]. PolyPublie.
<https://publications.polymtl.ca/58217/>

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

URL de PolyPublie: <https://publications.polymtl.ca/58217/>
PolyPublie URL:

Directeurs de recherche: Dominique Orban
Advisors:

Programme: Doctorat en mathématiques
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Méthodes de région de confiance pour l'optimisation non lisse

GEOFFROY LECONTE

Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Mathématiques

Avril 2024

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

Méthodes de région de confiance pour l'optimisation non lisse

présentée par **Geoffroy LECONTE**

en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Hanane DAGDOUGUI, présidente

Dominique ORBAN, membre et directeur de recherche

Youssef DIOUANE, membre

Alberto DE MARCHI, membre externe

DÉDICACE

À mes parents et à Louis

REMERCIEMENTS

Je tiens tout d’abord à remercier mon directeur de recherche Dominique Orban. Sa supervision m’a permis de travailler sur des sujets passionnants, de développer mon autonomie, mes compétences techniques et ma rigueur. Il m’aura aussi donné la chance de participer à des projets au sein de l’organisation JuliaSmoothOptimizers, qui aura été pour moi un formidable moyen de m’initier au développement de code libre d’accès en collaboration avec d’autres personnes talentueuses.

Je remercie le GERAD pour m’avoir accueilli durant cette thèse, et son personnel pour sa disponibilité et son aimabilité, IVADO pour m’avoir fourni une bourse lors de mon stage de fin d’études à l’École Centrale de Nantes m’ayant permis de m’installer à Montréal, et le département de Mathématiques et Génie Industriel de Polytechnique Montréal. Je souhaite aussi exprimer ma gratitude envers Hanane Dagdougui, Youssef Diouane, Alberto De Marchi et Jean-Philippe Harvey d’avoir accepté d’être membres de mon jury.

Merci à Sébastien Bourguignon pour son encadrement dans le cadre de l’option Recherche et Développement à l’École Centrale de Nantes, et pour m’avoir mis en contact avec Dominique afin de décrocher mon stage de fin d’études qui a donné lieu à cette thèse.

Je souhaite exprimer ma gratitude envers Tangi, pour sa bienveillance et son aide dans l’intégration de mes modules à l’organisation JuliaSmoothOptimizers, et Alexis, pour ses précieux conseils au début de mon doctorat, mais aussi pour les bons moments que nous avons eus l’occasion de partager en tant que colocataires et lors de nombreuses randonnées.

Une grosse pensée pour mes amis du beach volley : Qi Hao pour les Spikes et l’ambiance communauto, Adrien pour les soirées au Adri-Penthouse et les meilleurs thés de Montréal, et Lauriane pour ses histoires pas toujours très nettes et la transmission de sa passion pour la ville d’Ottawa. Également un énorme merci à mes colocataires, collègues, et coéquipiers : Paul, coach de renommée internationale spécialiste de compactage de mapreduce, et Romain, joueur d’échecs d’une sérénité remarquable toujours prêt à m’accompagner lors de mes manques de pizza.

Je remercie aussi les différents ensembles m’ayant permis d’exercer ma passion pour la musique : Polyjam, 9 à 5, et la Symphonie des Vents de Montréal.

Ce dernier paragraphe est dédié à ma famille. Je remercie tout d’abord mes parents pour leur soutien et pour m’avoir toujours poussé à donner le meilleur de moi-même. Merci à ma tante et marraine Bénédicte et à mon parrain Bernard pour leur transmission de leur passion pour

la recherche. Et enfin, merci à mon grand-oncle Louis pour mes nombreux passages à Évry et plus généralement pour tous les moments que nous avons passés ensemble, je regrette de n'avoir pas pu te revoir une dernière fois avant la fin de ma thèse.

RÉSUMÉ

Dans cette thèse, on s'intéresse aux méthodes de région de confiance utilisées pour la minimisation de la somme d'une fonction lisse f et d'une fonction non lisse h , éventuellement avec des contraintes d'inégalités. Nos contributions se basent sur un algorithme déjà existant, utilisant une méthode de région de confiance avec des approximations quasi-Newton appelée TR (*Trust Region* en anglais), et à laquelle nous apportons quelques corrections et clarifications. Nous développons une méthode de région de confiance inspirée de TR, mais utilisant des approximations quasi-Newton diagonales, appelée TRDH (*Trust Region with Diagonal Hessian approximations* en anglais). Cette méthode nous permet de résoudre le sous-problème de région de confiance de manière analytique, contrairement à TR qui utilise un algorithme itératif. TRDH peut également être utilisé comme sous-solveur de TR pour résoudre le sous-problème de région de confiance, à la place de l'algorithme de régularisation quadratique R2. L'analyse de convergence de TR et de TRDH est étendue dans le cas d'approximations de la matrice hessienne non bornées, sous réserve que la norme de ces approximations n'augmente pas trop vite. Nous obtenons une nouvelle borne de complexité, basée sur un paramètre contrôlant la croissance de la norme de ces approximations, qui est également applicable pour les problèmes lisses dans le cas où $h = 0$. Nous présentons une fonction f avec $h = 0$ pour laquelle notre nouvelle borne est atteinte. Nous élaborons une méthode de points intérieurs pour minimiser $f + h$ avec des contraintes de bornes, appelée RIPM (*Regularized Interior Proximal Method* en anglais). Notre algorithme résout une suite de problèmes barrière non contraints, dont les objectifs s'approchent de l'infini à proximité d'une borne du problème initial, en utilisant une méthode de région de confiance similaire à TR ou à TRDH. Nos algorithmes sont disponibles dans les modules Julia `RegularizedOptimization.jl` et `ShiftedProximalOperators.jl`. Nos tests sont effectués sur des problèmes du module `RegularizedProblems.jl`, et nous nous intéressons principalement à limiter le nombre d'évaluations d'objectifs et de gradients de f . Pour l'ensemble des problèmes testés, au moins une des variantes de TRDH ou de TR avec TRDH comme sous-solveur est plus efficace que R2 ou que TR avec R2 comme sous-solveur. Nous constatons que RIPM est le plus performant pour résoudre des problèmes avec contraintes de bornes ayant besoin de beaucoup d'itérations pour converger, et qu'il affiche des performances similaires à celles de TR, TRDH et R2 sur les autres problèmes avec contraintes de bornes.

ABSTRACT

This thesis focuses on trust-region methods used to minimize the sum of a smooth function f and a nonsmooth function h , possibly with inequality constraints. Our contributions are based upon an already existing algorithm that uses a quasi-Newton trust-region method named TR (*Trust Region*), to which we provide some corrections and clarifications. Utilizing TR as a foundation, we develop a new quasi-Newton trust-region method with diagonal Hessian approximations named TRDH (*Trust Region with Diagonal Hessian approximations*). With this method, we dispose from an analytical expression of a solution of the trust-region subproblem, unlike TR, which requires the use of an iterative solver. TRDH can also be used as a subproblem solver inside TR to solve the trust-region subproblem, instead of a quadratic regularization solver named R2 that is used by default. The convergence analysis of TR and TRDH is extended to unbounded Hessian approximations, provided that those approximations do not grow too quickly. We obtain a new complexity bound that is based upon a parameter controlling the growth of the Hessian approximations. This bound is also valid for smooth problems where $h = 0$. We show that our bound is sharp by constructing a function f for which it is attained when $h = 0$. We present an interior-point method used to minimize $f + h$ with inequality constraints, named RIPM (*Regularized Interior Proximal Method*). Our algorithm uses TR or TRDH to solve a sequence of unconstrained barrier subproblems, whose objectives approach infinity near a bound of the initial problem. Our algorithms are available from the Julia packages `RegularizedOptimization.jl` and `ShiftedProximalOperators.jl`. Our tests are performed on problems of the package `RegularizedProblems.jl`, and our goal is primarily to reduce the number of objective and gradient evaluations of f . For all the tested problems, at least one variant of TRDH or TR with TRDH as subsolver performs better than R2 or TR with R2 as subsolver. We observe that RIPM is the most efficient solver on bound-constrained problems that require many iterations to converge, and that it performs similarly to TR, TRDH, and R2 on the other bound-constrained problems.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	vi
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Objectifs de recherche	2
1.2 Plan de la thèse et contributions apportées	3
CHAPITRE 2 REVUE DE LITTÉRATURE	4
2.1 Définitions générales	4
2.2 Optimisation sans contraintes	8
2.3 Méthodes quasi-Newton	9
2.3.1 BFGS et L-BFGS	10
2.3.2 SR1 et L-SR1	11
2.3.3 Une méthode de régularisation quadratique	11
2.4 Méthodes de région de confiance	12
2.4.1 Sous-problème quadratique	13
2.4.2 Point de Cauchy	14
2.5 Optimisation non lisse	15
2.5.1 Outils d'analyse non lisse	16
2.5.2 Méthode du gradient proximal	19
2.5.3 Extensions	21
2.5.4 Une méthode de région de confiance proximale utilisant des approxima- tions quasi-Newton	22
2.5.5 Une méthode proximale de régularisation quadratique	24

2.6	Méthodes de points intérieurs	25
2.6.1	Problèmes barrières pour les contraintes d'inégalités	25
2.6.2	Méthode de région de confiance pour les problèmes barrières	26
CHAPITRE 3 CORRIGENDUM: A PROXIMAL QUASI-NEWTON TRUST-REGION METHOD FOR NONSMOOTH REGULARIZED OPTIMIZATION		30
CHAPITRE 4 ARTICLE 1: THE INDEFINITE PROXIMAL GRADIENT METHOD		35
4.1	Introduction	36
4.2	Background	39
4.3	The indefinite proximal operator	42
4.3.1	Definition and Examples	42
4.3.2	Properties	44
4.4	Diagonal quasi-Newton methods	46
4.5	The indefinite proximal gradient method	47
4.5.1	Description of the algorithm and convergence properties	47
4.5.2	A variant saving proximal operator computations	52
4.6	Implementation and numerical experiments	57
4.6.1	Basis pursuit denoise (BPDN)	59
4.6.2	Sparse nonnegative matrix factorization (NNMF)	60
4.6.3	Nonlinear support vector machine (SVM)	62
4.6.4	FitzHugh-Nagumo inverse problem	64
4.7	Discussion and future work	65
CHAPITRE 5 ARTICLE 2: COMPLEXITY OF TRUST-REGION METHODS WITH UNBOUNDED HESSIAN APPROXIMATIONS FOR SMOOTH AND NON- SMOOTH OPTIMIZATION		69
5.1	Introduction	70
5.2	Context	73
5.2.1	Background	73
5.2.2	Models and trust-region algorithm	75
5.3	Convergence and complexity with potentially unbounded Hessian	78
5.4	Sharpness of the complexity bound	83
5.5	Numerical verification of the bound	90
5.6	Discussion	92

CHAPITRE 6	ARTICLE 3: AN INTERIOR-POINT TRUST-REGION METHOD FOR NONSMOOTH REGULARIZED BOUND-CONSTRAINED OPTIMIZATION	94
6.1	Introduction	95
6.2	Background	98
6.3	Stationarity	101
6.4	Projected-directions methods	104
6.5	Barrier methods	105
6.5.1	Barrier subproblem	106
6.5.2	Convergence of the inner iterations	110
6.5.3	Convergence of the outer iterations	116
6.5.4	Convergence with a new criticality measure	123
6.6	Implementation and numerical experiments	129
6.6.1	Box-constrained quadratic problem	131
6.6.2	Sparse nonnegative matrix factorization (NNMF)	132
6.6.3	FitzHugh-Nagumo problem (FH)	134
6.6.4	Constrained basis pursuit denoise (BPDN)	135
6.7	Discussion and future work	136
CHAPITRE 7	CONCLUSION	139
7.1	Synthèse des travaux	139
7.2	Limitations de la solution proposée	140
7.3	Améliorations futures	140
RÉFÉRENCES		142

LISTE DES TABLEAUX

Table 4.1	BPDN (4.36) statistics with $h = \lambda\ \cdot\ _0$. All variants of TR use an LSR1 Hessian approximation, and are given a maximum of 100 inner iterations. The optimal objective value is $f(x_\star) = 9.90e - 03$	59
Table 4.2	Constrained BPDN (6.87) statistics with $h = \lambda\ \cdot\ _0$. All variants of TR use an LSR1 Hessian approximation, and are given a maximum of 100 inner iterations. The optimal objective value is $f(x_\star) = 8.98e - 03$	61
Table 4.3	NNMF (6.84) statistics with $h = \lambda\ \cdot\ _0$. All variants of TR use an LSR1 Hessian approximation, and are given a maximum of 100 inner iterations.	61
Table 4.4	SVM (4.39) statistics with $h = \lambda\ \cdot\ _1$. All variants of TR use an LBFGS Hessian approximation, and are given a maximum of 100 inner iterations. The train and test accuracies are the percentage of images correctly classified, computed by counting the number of elements of the residual $\mathbf{1} - \tanh(b \odot (A^T x))$ that are lower than 1 for the train and test problems respectively.	64
Table 4.5	FH (4.41) statistics with $h = \lambda\ \cdot\ _0$. All variants of TR use an LBFGS Hessian approximation, and are given a maximum of 200 inner iterations. The optimal objective value is $f(x_\star) = 1.03e + 00$	66
Table 4.6	FH (4.41) (left) and constrained FH (6.86) (right) solutions identified by the solvers tested. The unconstrained solution is $(0, 0.2, 1, 0, 0)$	67
Tableau 4.7	Constrained FH (6.86) statistics with $h = \lambda\ \cdot\ _1$. All variants of TR use an LBFGS Hessian approximation, and are given a maximum of 200 inner iterations.	68
Table 5.1	Rounded theoretical values of $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ for $\epsilon = 1/3$	91
Table 6.1	Statistics of (6.83). TR and RIPM use an LSR1 Hessian approximation. The maximum number of objective evaluations is set to 800.	132
Table 6.2	Statistics of (6.84). TR and RIPM use an LSR1 Hessian approximation. The maximum number of objective evaluations is set to 8000.	134
Table 6.3	Statistics of (6.86). TR and RIPM use an LBFGS Hessian approximation.	134
Table 6.4	Statistics of (6.87). TR and RIPM use an LSR1 Hessian approximation.	137

LISTE DES FIGURES

Figure 4.1	Illustration of Example 1 with $\delta_i > 0$ (left), $\delta_i = 0$ (center), and $\delta_i < 0$ (right). The blue curve is $g_i x_i + \frac{1}{2} \delta_i x_i^2$ while the red curve is $g_i x_i + \frac{1}{2} \delta_i x_i^2 + \lambda x_i _0$	43
Figure 4.2	Illustration of Example 2. Top row: $\delta_i = 0$. From left to right: $0 \leq g_i < \lambda$, $g_i = \lambda$, and $g_i > \lambda$. Middle row: $\delta_i > 0$. From left to right: $g_i < -\lambda$, $ g_i \leq \lambda$, and $g_i > \lambda$. Bottom row: $\delta_i < 0$. From left to right: $g_i \leq -\lambda$, $ g_i < \lambda$, and $g_i > \lambda$	45
Figure 4.3	Solution of (4.36) (left), and error (right) with TRDH-Spec.	60
Figure 4.4	Solution of (6.87) (left), and error (right) with TR-TRDH-Spec. . . .	62
Figure 4.5	Solution of (6.84) with TR-TRDH-PSB.	63
Figure 4.6	Solution map of (4.39) with TR-TRDH-Spec (left), and sample 1's (right, top row) and 7's (right, bottom row) rotated digits from the MNIST dataset.	65
Figure 4.7	Solution of (4.41) (left) and of (6.86) (right) with TR-TRDH-PSB and sampled values of V and W	68
Figure 5.1	TR logs with $\epsilon = 1/3$. <i>outer</i> denotes the iteration number, <i>inner</i> is the number of iterations performed by the subsolver to solve (5.15) with the model in (5.11), $\sqrt{\xi_{cp}}/\sqrt{\nu}$ is $\nu_k^{-1/2} \xi_{cp}(\Delta_k; x_k, \nu_k)^{1/2}$, $\sqrt{\xi}$ is the numerator of (5.16), $\ s\ $ is $\ s_k\ $, and the remaining columns refer unambiguously to data used in Algorithm 5.2.1.	91
Figure 5.2	Illustration of example (5.51) with $\epsilon = 1/3$. Top row : values of f (left) and of f' (right) for $x \in [0, x_{k_\epsilon+1}]$. Bottom row : iterates x_k (left) and steps s_k (right) for $k \in [0, k_\epsilon + 1]$	93
Figure 6.1	Plots of the objective of (6.83) per gradient evaluation with different solvers.	133
Figure 6.2	Plots of the objective of (6.84) per gradient evaluation with different solvers.	135
Figure 6.3	Plots of the objective of (6.86) per gradient evaluation with different solvers.	136
Figure 6.4	Plots of the objective of (6.87) per gradient evaluation with different solvers.	138

CHAPITRE 1 INTRODUCTION

L'optimisation est une branche des mathématiques ayant de nombreuses applications dans des domaines tels que l'ingénierie, l'industrie, l'économie, etc. À partir des années 1940, le développement des plateformes de calcul et d'algorithmes tels que la méthode du simplexe de George Dantzig pour les problèmes linéaires permettent de résoudre des problèmes de plus en plus grands, et suscitent l'intérêt des chercheurs. Par la suite, de nombreux autres algorithmes pour l'optimisation adaptés à des problèmes ayant différentes structures seront découverts.

Un exemple de problème d'optimisation est

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

où la fonction objectif f est une fonction lisse, c'est-à-dire qu'elle est différentiable et que ses dérivées sont continues sur $\mathbb{R}^n$. Il existe plusieurs types d'algorithmes pour résoudre ce problème, dont les méthodes de région de confiance, qui auront un rôle crucial dans cette thèse. Ces méthodes se basent sur la résolution à chaque itération k d'un sous-problème dont l'objectif est un modèle simple m_k de f à l'intérieur d'une région centrée en l'itérée actuelle, appelée région de confiance. Si la solution de ce sous-problème permet de faire diminuer suffisamment l'objectif f , l'itération suivante aura tendance à augmenter la taille de la région de confiance. Au contraire, si l'objectif ne diminue pas suffisamment, la taille de la région de confiance est réduite. Le modèle m_k utilise souvent des informations sur les dérivées de f , l'existence de ces dernières est donc importante pour les méthodes de région de confiance.

Les problèmes d'optimisation peuvent également avoir des contraintes. Par exemple,

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.c.} \quad \ell \leq x \leq u,$$

où $\ell \leq u$, est un problème d'optimisation avec contraintes d'inégalités. Les méthodes de points intérieurs, développées à partir des années 1960, sont souvent utilisées pour résoudre ces problèmes. Elles consistent à résoudre une suite de sous-problèmes non contraints

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

où la fonction barrière ϕ_k tend vers l'infini à l'approche d'une borne. Sous certaines hypothèses, la suite des solutions des sous-problèmes barrière converge vers une solution du problème contraint.

Dans cette thèse, nous considérons un type de problème plus général ayant la forme

$$\min_{x \in \mathbb{R}^n} f(x) + h(x), \quad \ell \leq x \leq u,$$

où $\ell_i \in \mathbb{R} \cup \{-\infty\}$ et $u_i \in \mathbb{R} \cup \{+\infty\}$ pour tout $i \in \{1, \dots, n\}$, f est lisse, mais h est non lisse. Ce type de problème est parfois appelé problème régularisé, car il est particulièrement utile pour résoudre des problèmes d'optimisation où h remplit le rôle de fonction de régularisation ayant pour but de promouvoir certaines propriétés (telles que le nombre d'éléments nuls) pour la solution x^* . Le choix $\ell_i = -\infty$ et $u_i = +\infty$ pour tout $i \in \{1, \dots, n\}$ permet de traiter le cas particulier non contraint

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

Pour certaines fonctions h dont l'opérateur proximal

$$\text{prox}_{\nu h}(x) = \arg \min_w \frac{1}{2} \nu^{-1} \|w - x\|^2 + h(w),$$

où $\nu > 0$, peut être calculé facilement, il est possible de résoudre ces problèmes d'optimisation non lisse par différentes méthodes. La méthode du gradient proximal est probablement la plus connue, mais il est aussi possible d'utiliser une méthode de région de confiance, tout en ayant une borne de complexité similaire à celle connue pour l'optimisation lisse.

1.1 Objectifs de recherche

L'article [4] présente une méthode de région de confiance pour l'optimisation non lisse pour laquelle les sous-problèmes sont basés sur un modèle quasi-Newton de f . Ces sous-problèmes ne peuvent en général pas être résolus de manière analytique et doivent être résolus de manière itérative, par exemple en utilisant la méthode du gradient proximal. De cette observation, nous dégagons notre premier objectif, consistant à implémenter un algorithme de région de confiance sans avoir besoin d'utiliser un sous-solveur itératif.

Si la suite des approximations quasi-Newton n'est pas bornée, la convergence de TR n'est pas garantie, contrairement au cas lisse (sous réserve que la norme des approximations quasi-Newton n'augmente pas trop vite). Ainsi, notre deuxième objectif a pour but de démontrer la convergence dans le cas d'approximations quasi-Newton non bornées.

Notre troisième objectif consiste à implémenter un algorithme de points intérieurs pour l'optimisation non lisse, pouvant notamment utiliser TR ou l'algorithme du premier objectif pour résoudre les sous-problèmes barrière.

Enfin, notre quatrième objectif est d'ajouter nos algorithmes au module `Julia RegularizedOptimization.jl` afin de pouvoir comparer leurs performances, ainsi que de les rendre facilement utilisables pour d'autres utilisateurs.

1.2 Plan de la thèse et contributions apportées

Le chapitre 2 établit une revue de littérature sur diverses techniques d'optimisation lisse et non lisse, incluant l'article [4], qui décrit une méthode de région de confiance pour l'optimisation non lisse, auquel nous apportons des corrections dans le chapitre 3.

Le chapitre 4 présente une méthode de région de confiance basée sur une approximation diagonale de la matrice des dérivées secondes. Cette approximation permet d'obtenir une solution analytique des sous-problèmes pour certaines fonctions h non lisses, contrairement à l'algorithme TR [4] qui résout les sous-problèmes de manière itérative.

Le chapitre 5 établit une borne de complexité pour TR dans le cas d'approximations de la matrice des dérivées secondes non bornées. Cette borne s'applique notamment aux problèmes lisses, et à l'algorithme avec approximation diagonale de la matrice des dérivées secondes mentionné plus haut.

Le chapitre 6 présente une méthode de points intérieurs pour résoudre les problèmes d'optimisation non lisse avec contraintes d'inégalités, en résolvant les sous-problèmes barrières avec une méthode de région de confiance similaire à celle décrite dans [4].

Enfin, le chapitre 7 résume les contributions apportées, établit les limites de ces contributions, et présente des idées d'améliorations futures.

Les algorithmes développés sont disponibles dans les modules `Julia RegularizedOptimization.jl` et `ShiftedProximalOperators.jl`. Ces modules font partie de l'organisation pour l'optimisation et l'algèbre linéaire numérique `JuliaSmoothOptimizers`, à laquelle de nombreuses contributions ont été réalisées durant cette thèse.

CHAPITRE 2 REVUE DE LITTÉRATURE

2.1 Définitions générales

Cette section vise à rappeler certains concepts utiles en optimisation. Dans le cas où le lecteur rencontrerait des termes avec lesquels il n'est pas familier, nous recommandons l'ouvrage [68], notamment l'annexe A et les chapitres 1 et 2.

Nous nous intéressons tout d'abord au problème d'optimisation continue

$$\min_x f(x) \quad \text{s.c.} \quad c(x) = 0, \quad g(x) \geq 0, \quad (2.1)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$, et $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ sont continûment différentiables. Le *domaine admissible* est défini comme l'ensemble des points pour lesquels les contraintes de (2.1) sont satisfaites :

$$\Omega = \{x \mid c(x) = 0, \quad g(x) \geq 0\}. \quad (2.2)$$

On dit que le problème (2.1) est *réalisable* s'il existe au moins un $x \in \Omega$. Le problème (2.1) est dit *non-borné* s'il existe une suite $\{x_k\}$ d'éléments de Ω telle que $f(x_k) \rightarrow -\infty$.

Les outils qui vont suivre serviront à établir des conditions de qualification pour les contraintes de (2.1), permettant de s'assurer que les contraintes sont définies correctement pour pouvoir trouver un point stationnaire. Nous adoptons une approche géométrique telle que celle présentée dans [68, Chapitre 12.7].

Nous rappelons qu'un ensemble $C \subset \mathbb{R}^n$ est un *cône* si $0 \in C$ et si $\lambda x \in C$ pour tout $x \in C$ et $\lambda > 0$.

Définition 1. Le cône tangent à Ω en $\bar{x}$ est l'ensemble

$$T_\Omega(\bar{x}) = \left\{ d \mid \exists \{x_k\} \subset \Omega, t_k \rightarrow 0, \frac{x_k - \bar{x}}{t_k} \rightarrow d \right\}. \quad (2.3)$$

Tout vecteur $d \in T_\Omega(\bar{x})$ est appelé vecteur tangent à Ω en $\bar{x}$.

On peut facilement vérifier que le cône tangent est bien un cône.

Définition 2. Le cône normal régulier à Ω en $\bar{x} \in \Omega$ est l'ensemble

$$\hat{N}_\Omega(\bar{x}) = \{v \mid \forall x \in \Omega, v^T(x - \bar{x}) \leq o(\|x - \bar{x}\|)\}. \quad (2.4)$$

Tout vecteur $v \in \hat{N}_\Omega(\bar{x})$ est appelé vecteur normal régulier à Ω en $\bar{x}$. Le cône normal (général) à Ω en $\bar{x} \in \Omega$ est l'ensemble

$$N_\Omega(\bar{x}) = \{v \mid \exists \{x_k\} \subset \Omega, x_k \rightarrow \bar{x}, \exists \{v_k\}, v_k \in \hat{N}_\Omega(x_k), v_k \rightarrow v\}. \quad (2.5)$$

Tout vecteur $v \in N_\Omega(\bar{x})$ est appelé vecteur normal (général) à Ω en $\bar{x}$.

Nous énonçons maintenant quelques propriétés sur les cônes normaux réguliers et généraux.

Proposition 1 (73, Proposition 6.5 et Théorème 6.9). *Soit $x \in \Omega$.*

1. $\hat{N}_\Omega(\bar{x})$ et $N_\Omega(\bar{x})$ sont tous les deux des cônes.
2. $\hat{N}_\Omega(\bar{x}) = \{v \mid \forall d \in T_\Omega(\bar{x}), v^T d \leq 0\}$.
3. $\hat{N}_\Omega(\bar{x}) \subset N_\Omega(\bar{x})$.
4. Si Ω est un ensemble convexe, $N_\Omega(\bar{x}) = \hat{N}_\Omega(\bar{x}) = \{v \mid \forall x \in \Omega, v^T(x - \bar{x}) \leq 0\}$.

Le théorème suivant nous donne une condition nécessaire du premier ordre pour trouver un minimum local.

Théorème 1 (73, Théorème 6.12). *Si $\bar{x}$ est un minimum local de (2.1), alors*

$$-\nabla f(\bar{x}) \in \hat{N}_\Omega(\bar{x}). \quad (2.6)$$

On note que la réciproque n'est pas forcément vraie sans hypothèses supplémentaires. Dans le cas non contraint, le cône normal est réduit au vecteur nul, et il suffit donc de résoudre $\nabla f(x) = 0$ pour vérifier la condition nécessaire du premier ordre.

Définition 3. *L'ensemble des contraintes actives en un point x réalisable est l'ensemble*

$$\mathcal{A}(x) := \{i \mid g_i(x) = 0\}. \quad (2.7)$$

L'ensemble des contraintes inactives en un point x réalisable est l'ensemble

$$\mathcal{I}(x) := \{i \mid g_i(x) > 0\}. \quad (2.8)$$

Définition 4. *Pour un point x réalisable, l'ensemble des directions réalisables linéarisées est défini par*

$$\mathcal{F}(x) := \{d \mid \forall i, d^T \nabla c_i(x) = 0, \forall i \in \mathcal{A}(x), d^T \nabla g_i(x) \geq 0\}. \quad (2.9)$$

Les contraintes de qualification en un point $x \in \Omega$ sont souvent définies comme des conditions qui s'assurent que $\mathcal{F}(x) = \mathcal{T}_\Omega(x)$.

Un exemple de contrainte de qualification est donné dans la définition suivante.

Définition 5 (LICQ). *Pour $x \in \Omega$, on dit que la condition d'indépendance linéaire (LICQ en anglais pour Linear Independence Constraint Qualification) est satisfaite si les vecteurs $\nabla c_i(x)$ et $\nabla g_j(x)$ pour $j \in \mathcal{A}(x)$ sont linéairement indépendants.*

Nous allons maintenant introduire des conditions nécessaires du premier ordre pour résoudre (2.1). Pour cela, nous définissons le *lagrangien*

$$\mathcal{L}(x, y, s) := f(x) + \sum_i y_i c_i(x) - \sum_j s_j g_j(x). \quad (2.10)$$

Le théorème suivant est connu sous le nom de théorème de *Karush-Kuhn-Tucker* (KKT).

Théorème 2 (68, Théorème 12.1). *Supposons que x^* est un minimum local de (2.1), et que la LICQ est vérifiée en x^* . Alors, il existe des multiplicateurs de Lagrange y^* et s^* tels que (x^*, y^*, s^*) vérifie les conditions de KKT*

$$\nabla_x \mathcal{L}(x^*, y^*, s^*) = 0, \quad (2.11a)$$

$$c(x^*) = 0, \quad (2.11b)$$

$$g(x^*) \geq 0, \quad (2.11c)$$

$$s^* \geq 0, \quad (2.11d)$$

$$s_i^* g_i(x^*) = 0. \quad (2.11e)$$

Cependant, lorsqu'on trouve (x, y, s) vérifiant les conditions (2.11), il n'est pas possible de savoir directement si x est un minimum local sans hypothèses supplémentaires sur (2.1). En fait, on pourrait avoir affaire à un maximum local, un minimum local, ou un point de selle (point qui n'est ni un minimum local, ni un maximum local). Pour savoir, il faut utiliser l'information du deuxième ordre sur le lagrangien.

Pour cela, nous définissons le *cône critique* $\mathcal{C}(x^*, y^*, s^*)$, tel que

$$w \in \mathcal{C}(x^*, y^*, s^*) \equiv \begin{cases} \nabla c_i(x^*)^T w = 0 \quad \forall i, \\ \nabla g_i(x^*)^T w = 0 \quad \forall i \in \mathcal{A}(x^*) \text{ with } s_i^* > 0, \\ \nabla g_i(x^*)^T w \geq 0 \quad \forall i \in \mathcal{A}(x^*) \text{ with } s_i^* = 0. \end{cases} \quad (2.12)$$

Le théorème suivant donne des conditions nécessaires du deuxième ordre.

Théorème 3 (68, Théorème 12.5). *Supposons que x^* est un minimum local de (2.1), et que la LICQ est vérifiée en x^* . Soit y^* et s^* les multiplicateurs de Lagrange pour lesquels les conditions de KKT sont vérifiées. Alors*

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, y^*, s^*) w \geq 0, \quad \forall w \in \mathcal{C}(x^*, y^*, s^*). \quad (2.13)$$

Nous disposons également d'une condition suffisante du deuxième ordre avec le théorème suivant.

Théorème 4 (68, Théorème 12.6). *Soit x^* un point réalisable. S'il existe y^* et s^* des multiplicateurs de Lagrange pour lesquels les conditions de KKT sont vérifiées, et si*

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, y^*, s^*) w > 0, \quad \forall w \in \mathcal{C}(x^*, y^*, s^*), w \neq 0, \quad (2.14)$$

alors x^* est un minimum local de (2.1).

En pratique, on pourra donc vérifier qu'un point satisfaisant les conditions de KKT données par le théorème 2 est un minimum local en utilisant le théorème 4. Dans beaucoup d'algorithmes, on requiert que le hessien du lagrangien soit défini positif en ce point, ainsi le théorème 4 peut s'appliquer.

Généralement, un problème d'optimisation tel que (2.1) ne peut pas être résolu directement, et requiert la génération d'itérées $x_k \rightarrow \bar{x}$ successives afin d'obtenir une solution approchée.

Enfin, nous rappelons le concept de vitesse de convergence d'une suite dans la définition suivante.

Définition 6. *Soit $\{x_k\}$ une suite telle que $x_k \rightarrow x^*$.*

1. *S'il existe $r > 0$ et*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow r, \quad (2.15)$$

on dit que $\{x_k\}$ converge linéairement vers x^* ,

2. *Si*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \rightarrow 0, \quad (2.16)$$

on dit que $\{x_k\}$ converge superlinéairement vers x^* .

3. *S'il existe $r > 0$ et*

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \rightarrow r, \quad (2.17)$$

on dit que $\{x_k\}$ converge quadratiquement vers x^* .

2.2 Optimisation sans contraintes

Pour le reste de cette section, nous cherchons à résoudre le problème sans contraintes

$$\min_x f(x). \quad (2.18)$$

Pour tout $k \in \mathbb{N}$, $x_{k+1} = x_k + t_k d_k$, où $t_k > 0$ et d_k est une direction de recherche. Beaucoup d'algorithmes requièrent la décroissance de l'objectif $f(x_{k+1}) < f(x_k)$ afin de faire un progrès vers le minimum.

Une *direction de descente* est une direction d_k telle que $\nabla f(x_k)^T d_k < 0$.

En pratique, on peut utiliser différentes stratégies pour le calcul du pas t_k . La *condition d'Armijo* est une condition bien connue permettant de s'assurer que l'objectif décroît suffisamment :

$$f(x_k + t_k d_k) \leq f(x_k) + c_1 t_k \nabla f(x_k)^T d_k, \quad (2.19)$$

où $c_1 \in (0, 1)$. Pour s'assurer que la taille du pas n'est pas trop petite, on peut choisir un t_k vérifiant la *condition de courbure*

$$\nabla f(x_k + t_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k, \quad (2.20)$$

avec $c_2 \in (c_1, 1)$. Ces deux conditions réunies forment les *conditions de Wolfe*. Le théorème suivant nous donne un résultat de complexité pour une suite d'itérées générée à partir de directions de descente et de pas t_k satisfaisant la condition de Wolfe.

Théorème 5 (68, Théorème 3.6). *Soit f deux fois continûment différentiable. On considère les itérés $x_{k+1} = x_k + t_k d_k$ où d_k est une direction de descente et t_k satisfait les conditions de Wolfe (2.19) et (2.20) avec $c_1 \leq 1/2$. Si $\{x_k\}$ converge vers x^* tel que $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est définie positive, et si la direction de recherche satisfait la condition de Dennis-Moré*

$$\frac{\|\nabla f(x_k) + \nabla^2 f(x_k) d_k\|}{\|d_k\|} \rightarrow 0, \quad (2.21)$$

alors

1. à partir d'un certain rang k_0 , le pas $t_k = 1$ est admissible pour tout $k > k_0$,
2. si $t_k = 1$ pour tout $k > k_0$, $\{x_k\}$ converge vers x^* superlinéairement.

Pour plus d'informations sur les techniques de recherche linéaire, voir [68, Chapitre 3].

Un exemple de direction de descente est $d_k = -\nabla f(x_k)$, et permet d'écrire Algorithme 2.2.1. La vitesse de convergence de cette méthode est souvent approximativement linéaire [19,

Chapitre 9.3]. On note que la condition d'arrêt est basée sur la norme du gradient afin de détecter un point stationnaire. Nous utilisons une tolérance absolue ϵ_a et une tolérance relative ϵ_r par rapport à la norme initiale du gradient.

Algorithme 2.2.1 Méthode de plus forte pente ou algorithme du gradient.

- 1: Choisir un point initial x_0 .
 - 2: Si $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, renvoyer x_k .
 - 3: $d_k = -\nabla f(x_k)$.
 - 4: Pour un t_k bien choisi, $x_{k+1} = x_k + t_k d_k$.
-

L'algorithme 2.2.2 est un autre algorithme pour l'optimisation sans contraintes, mais il requiert de connaître le hessien de f , qui n'existe pas forcément avec les hypothèses de (2.1), et qui peut être coûteux à calculer.

Algorithme 2.2.2 Méthode de Newton.

- 1: Choisir un point initial x_0 .
 - 2: Si $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, renvoyer x_k .
 - 3: Résoudre $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$ pour trouver d_k .
 - 4: Pour un t_k bien choisi, $x_{k+1} = x_k + t_k d_k$.
-

La convergence de cette méthode est souvent plus rapide que celle du gradient, et est quadratique à l'approche d'un point stationnaire [19, Chapitre 9.5]. Cependant, si $\nabla^2 f(x_k)$ n'est pas définie positive, la direction d_k calculée pourrait ne pas être une direction de descente.

Si le hessien n'est pas disponible ou est trop coûteux à calculer, il est possible d'utiliser une méthode *quasi-Newton* telle que l'algorithme 2.2.3. Ces méthodes utilisent une approximation B_k de $\nabla^2 f(x_k)$ (sous réserve d'existence), que l'on peut forcer à être définie positive.

Algorithme 2.2.3 Méthode quasi-Newton.

- 1: Choisir un point initial x_0 .
 - 2: Si $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, renvoyer x_k .
 - 3: Choisir $B_k = B_k^T$, $B_k \succ 0$, et résoudre $B_k d_k = -\nabla f(x_k)$ pour trouver d_k .
 - 4: Pour un t_k bien choisi, $x_{k+1} = x_k + t_k d_k$.
-

2.3 Méthodes quasi-Newton

Dans cette sous-section, nous présenterons deux approximations quasi-Newton qui seront utilisées dans les chapitres suivants. Nous basons notre présentation sur [68, Chapitre 6]. Il en

existe d'autres, et notamment des approximations quasi-Newton diagonales, dont certaines seront rappelées dans le chapitre 5.

Une approximation quasi-Newton peut être utilisée pour construire un modèle de f à l'itération k :

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s. \quad (2.22)$$

On pose $s_k = t_{k-1} d_{k-1} = x_k - x_{k-1}$. Une idée pour calculer une approximation quasi-Newton est d'imposer que $\nabla m_k(0) = \nabla f(x_k)$ (ici déjà satisfaite au vu de l'expression de m_k) et $\nabla m_k(-s_k) = \nabla f(x_k) - B_k s_k = \nabla f(x_{k-1})$. Cette dernière condition s'appelle l'équation sécante, et se réécrit :

$$B_k s_k = y_k, \quad (2.23)$$

où $y_k = \nabla f(x_k) - \nabla f(x_{k-1})$.

2.3.1 BFGS et L-BFGS

Pour déterminer B_k , la méthode BFGS (pour le nom de ses auteurs : Broyden, Fletcher, Goldfarb, Shanno) résout le problème suivant :

$$\min_B \|B - B_{k-1}\| \quad \text{s.c.} \quad B = B^T, \quad B s_k = y_k, \quad (2.24)$$

pour une certaine norme matricielle. Par exemple, il est possible d'utiliser $\|B\|_W = \|W^{1/2} B W^{1/2}\|_F$, pour une certaine matrice W qui ne sera pas détaillée dans cette thèse. Pour plus de détails, voir [68, Chapitre 6.1].

Cela mène à la formule suivante pour la mise à jour :

$$B_k = (I - \rho_k y_k s_k^T) B_{k-1} (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T, \quad (2.25)$$

où $\rho_k = 1/y_k^T s_k$.

Il peut être préférable d'obtenir une formule pour l'inverse de l'approximation quasi-Newton $H_k := B_k^{-1}$. Pour cela, on peut résoudre le problème

$$\min_H \|H - H_{k-1}\| \quad \text{s.c.} \quad H = H^T, \quad H y_k = s_k. \quad (2.26)$$

En utilisant la même norme, la formule de mise à jour s'écrit

$$H_k = (I - \rho_k s_k y_k^T) H_{k-1} (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T. \quad (2.27)$$

Pour l'optimisation en grande dimension, le stockage des coefficients de la matrice H_k peut s'avérer être trop coûteux. Si on reprend (2.27) en posant $V_k = I - \rho_k y_k s_k^T$, on pourrait appliquer la mise à jour de manière récursive jusqu'à obtenir H_k en fonction de H_0 . En pratique, plus k est grand et plus cela va demander de stocker un grand nombre de vecteurs. On préfère plutôt choisir un petit entier m et calculer une version de BFGS à mémoire limitée (L-BFGS)

$$\begin{aligned}
H_k = & (V_k^T \dots V_{k-m+1}^T) H_k^0 (V_{k-m+1} \dots V_k) \\
& + \rho_{k-m+1} (V_k^T \dots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \dots V_k) \\
& + \dots \\
& + \rho_{k-1} V_k^T s_{k-1} s_{k-1}^T V_k \\
& + \rho_k s_k s_k^T,
\end{aligned} \tag{2.28}$$

où H_k^0 est une approximation initiale donnée par l'utilisateur.

2.3.2 SR1 et L-SR1

La mise à jour symétrique de rang 1 (SR1, voir [68, Chapitre 6.2]) prend la forme

$$B_k = B_{k-1} + \sigma v v^T, \tag{2.29}$$

où $\sigma = \pm 1$, et v est un vecteur. Avec la condition additionnelle que B_k satisfasse l'équation sécante, la mise à jour est (voir)

$$B_k = B_{k-1} + \frac{(y_k - B_{k-1} s_k)(y_k - B_{k-1} s_k)^T}{(y_k - B_{k-1} s_k)^T s_k}. \tag{2.30}$$

Pour la mise à jour inverse, la formule de Sherman-Morrison pour les mises à jour de rang 1 (voir [68, Annexe A.28]) permet d'obtenir

$$H_k = H_{k-1} + \frac{(s_k - H_{k-1} y_k)(s_k - H_{k-1} y_k)^T}{(s_k - H_{k-1} y_k)^T y_k}. \tag{2.31}$$

2.3.3 Une méthode de régularisation quadratique

L'algorithme de régularisation quadratique (R2) ou algorithme de régularisation adaptative (appelé AR1 dans [30, Chapitre 2.4]), que nous réécrivons en tant qu'algorithme 2.3.1, utilise un modèle de Taylor d'ordre 1 accompagné du terme de régularisation quadratique de la forme

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} \sigma_k \|s\|^2, \tag{2.32}$$

où $\sigma_k > 0$. Avec un tel modèle, [30, Chapitre 2.4] montre que l'algorithme 2.3.1 met $O(\epsilon^{-2})$ itérations pour atteindre $\nabla f(x_k) \leq \epsilon$.

En faisant l'identification $B_k = \sigma_k I$, on constate que l'algorithme 2.3.1 est une méthode quasi-Newton.

Algorithme 2.3.1 L'algorithme R2 [30, Algorithme 2.4.1].

1: Choisir les constantes

$$0 < \eta_1 \leq \eta_2 < 1, \quad \text{et} \quad 0 < \gamma_3 \leq 1 \leq \gamma_1 \leq \gamma_2.$$

2: Choisir $x_0 \in \mathbb{R}^n$, $\sigma_0 > 0$, calculer $f(x_0)$.

3: **pour** $k = 0, 1, \dots$ **faire**

4: Si $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, renvoyer x_k .

5: Définir un modèle m_k comme en (2.32) et calculer un point s_k .

6: Calculer le rapport

$$\rho_k := \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

7: Si $\rho_k \geq \eta_1$, $x_{k+1} = x_k + s_k$. Sinon, $x_{k+1} = x_k$.

8: Mettre à jour σ_k suivant

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

Il est aussi possible d'utiliser un autre modèle donnant naissance à une variante nommée ARC (Adaptative Regularization Cubics [27]) où le modèle considéré est

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s + \frac{1}{3} \sigma_k \|s\|^3. \quad (2.33)$$

On constate que la mise à jour de σ_k dans l'algorithme 2.3.1 contient beaucoup de points communs avec la mise à jour du rayon d'une méthode de région de confiance, que nous allons présenter dans la section qui suit.

2.4 Méthodes de région de confiance

Les méthodes de région de confiance consistent à faire des approximations successives de f à l'intérieur d'une boule dont le centre est l'itérée actuelle x_k . La direction de recherche s_k sera calculée à partir du modèle de f à l'itération k , et, en fonction de la valeur de $f(x_k + s_k)$, on va mettre à jour (ou non) l'itérée actuelle et le rayon de la boule utilisée. Le livre [34] développe

plus en détail les méthodes de région de confiance et leurs extensions. L'algorithme 2.4.1 est une méthode de région de confiance basique, inspirée de [34, Algorithme 6.1.1].

Algorithme 2.4.1 Méthode de région de confiance classique.

1: Choisir les constantes

$$0 < \eta_1 \leq \eta_2 < 1, \quad \text{et} \quad 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3 \leq \gamma_4.$$

2: Choisir $x_0 \in \mathbb{R}^n$, $\Delta_0 > 0$, calculer $f(x_0)$.

3: **pour** $k = 0, 1, \dots$ **faire**

4: Si $\|\nabla f(x_k)\| \leq \epsilon_a + \epsilon_r \|\nabla f(x_0)\|$, renvoyer x_k .

5: Définir un modèle m_k et calculer un point s_k tel que $x_k + s_k \in B(x_k, \Delta_k)$.

6: Calculer le rapport

$$\rho_k := \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

7: Si $\rho_k \geq \eta_1$, $x_{k+1} = x_k + s_k$. Sinon, $x_{k+1} = x_k$.

8: Mettre à jour Δ_k suivant

$$\Delta_{k+1} \in \begin{cases} [\gamma_3 \Delta_k, \gamma_4 \Delta_k] & \text{si } \rho_k \geq \eta_2, & \text{(itération très satisfaisante)} \\ [\gamma_2 \Delta_k, \Delta_k] & \text{si } \eta_1 \leq \rho_k < \eta_2, & \text{(itération satisfaisante)} \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{si } \rho_k < \eta_1 & \text{(itération non satisfaisante).} \end{cases}$$

C'est le rapport ρ_k qui va permettre de décider comment mettre à jour le rayon de notre région de confiance. S'il est grand, on aura tendance à augmenter ce rayon car le modèle choisi approxime bien f , tandis que s'il est petit, on va réduire la taille de la région de confiance pour tenter d'avoir une meilleure approximation locale de f avec notre modèle.

2.4.1 Sous-problème quadratique

Un exemple de modèle très utilisé est le modèle quadratique

$$m_k(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s, \quad (2.34)$$

où B_k est par exemple une approximation quasi-Newton de $\nabla^2 f(x_k)$.

Le problème de région de confiance à résoudre à chaque itération devient alors

$$\min_s f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s, \quad \text{s.c. } \|s\|_2 \leq \Delta_k. \quad (2.35)$$

Le théorème suivant, basé sur [63] et [45], nous donne un résultat sur la solution du problème de région de confiance avec un modèle quadratique.

Théorème 6 (68, Théorème 4.1). s_k est solution de (2.35) si et seulement si s_k est réalisable et s'il existe $\lambda \geq 0$ tel que

$$(B_k + \lambda I)s_k = -\nabla f(x_k), \quad (2.36a)$$

$$\lambda(\Delta_k - \|s_k\|_2) = 0, \quad (2.36b)$$

$$(B_k + \lambda I) \text{ est semi-définie positive.} \quad (2.36c)$$

Comme B_k est symétrique, il existe une matrice orthogonale U_k et une matrice diagonale Λ_k telles que $B_k + \lambda I = U_k^T(\Lambda_k + \lambda I)U_k$. Si $\lambda = 0$ satisfait (2.36), on peut trouver s_k facilement. Sinon, soit

$$s(\lambda) = -(B_k + \lambda I)^{-1}\nabla f(x_k), \quad (2.37)$$

avec λ assez grand pour que $(B_k + \lambda I)$ soit définie positive. Il faut trouver $\lambda > 0$ tel que $\|s(\lambda)\| = \Delta$. Résoudre cette équation peut s'avérer être difficile, en pratique,

$$\frac{1}{\Delta_k} - \frac{1}{\|s(\lambda)\|} = 0 \quad (2.38)$$

sera préférée, voir [34, Chapitre 7.3] et [68, Chapitre 4.3] pour plus de détails. Cela mène à l'algorithme 2.4.2 pour la résolution du sous-problème (2.35) et l'obtention de s_k .

Algorithme 2.4.2 Résolution du sous-problème de région de confiance quadratique [68, Algorithme 4.3].

- 1: Choisir $\lambda^{(0)}, \Delta > 0$
- 2: **pour** $j = 0, 1, \dots$ **faire**
- 3: Factoriser $B_k + \lambda^{(j)}I = R^T R$.
- 4: Résoudre $R^T R s_j = -\nabla f(x_k), R^T u_j = s_j$.
- 5: Mettre à jour

$$\lambda^{(j+1)} = \lambda^{(j)} + \left(\frac{\|s_j\|^2}{\|q_j\|} \right)^2 \left(\frac{\|s_j\| - \Delta}{\Delta} \right).$$

2.4.2 Point de Cauchy

Cette section se base encore une fois sur [68, Chapitre 4.1] et sur [34, Chapitre 6.3]. Dans un premier temps, nous calculons

$$s_k^S = \arg \min_s f(x_k) + \nabla f(x_k)^T s \quad \text{s.c.} \quad \|s_k\| \leq \Delta_k, \quad (2.39)$$

ainsi que

$$t_k = \arg \min_{t \geq 0} m_k(ts_k^S) \quad \text{s.c.} \quad \|ts_k^S\| \leq \Delta_k. \quad (2.40)$$

Le point de Cauchy s'écrit alors

$$s_k^C = t_k s_k^S. \quad (2.41)$$

Il est possible d'expliciter complètement sa formule de la manière suivante :

$$s_k^C = -t_k \frac{\Delta_k}{\|\nabla f(x_k)\|} \nabla f(x_k), \quad (2.42)$$

où

$$t_k = \begin{cases} 1 & \text{si } \nabla f(x_k)^T B_k \nabla f(x_k) \leq 0, \\ \min(\|\nabla f(x_k)\|^3 / (\Delta_k \nabla f(x_k)^T B_k \nabla f(x_k)), 1) & \text{sinon.} \end{cases} \quad (2.43)$$

Le but du point de Cauchy est d'obtenir une décroissance suffisante du modèle m_k qui permettra à l'algorithme de progresser, sans avoir à calculer la solution exacte du sous-problème à l'itération k . Dans beaucoup d'algorithmes de régions de confiance, le point de Cauchy est calculé, puis on tente de l'améliorer. Par exemple, dans [30, Chapitre 2.3], il est indiqué que l'algorithme 2.4.1 utilisant un pas s_k vérifiant $m_k(s_k) \leq m_k(s_k^C)$ avec m_k défini en (2.34) met $O(\epsilon^{-2})$ itérations pour atteindre $\|\nabla f(x_k)\| \leq \epsilon$.

2.5 Optimisation non lisse

Dans cette section, nous chercherons à résoudre des problèmes de la forme

$$\min_x f(x) + h(x), \quad (2.44)$$

où f est continûment différentiable, ∇f est Lipschitzienne de constante $L > 0$, et h est une fonction pouvant être non lisse et vérifiant certaines hypothèses qui seront explicitées par la suite. h est souvent une fonction de régularisation utilisée pour mettre en avant certaines propriétés de la solution, notamment son caractère creux (promouvoir un grand nombre de zéros). Par exemple, on pourrait choisir $h = \lambda \|\cdot\|_1$ pour un paramètre $\lambda > 0$.

Il existe d'autres types de problèmes en optimisation non lisse, par exemple des problèmes de la forme

$$\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^m} f(x) + h(z), \quad \text{s.c.} \quad Ax + Bz = c, \quad (2.45)$$

où $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, $c \in \mathbb{R}^p$, f et g sont convexes, propres et semi-continues inférieures.

Ce type de problème peut être résolu avec des méthodes ADMM (*Alternating Direction Method of Multipliers* en anglais) [18], mais ne sera pas détaillé dans ce document.

2.5.1 Outils d'analyse non lisse

Dans cette sous-section, nous allons présenter quelques outils d'analyse importants en optimisation non lisse. Pour les définitions suivantes, nous considérons une fonction $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$, où $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

Définition 7. Une fonction f est dite propre s'il existe au moins un $x \in \mathbb{R}^n$ tel que $f(x) < +\infty$, et pour tout $x \in \mathbb{R}^n$, $f(x) > -\infty$.

Nous rappelons maintenant les concepts de limite inférieure et de semi-continuité inférieure.

Définition 8 (73, Définition 1.5). La limite inférieure d'une fonction $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ en $\bar{x}$ est définie par

$$\liminf_{x \rightarrow \bar{x}} f(x) := \lim_{\delta \searrow 0} \left(\inf_{x \in B(\bar{x}, \delta)} f(x) \right) = \sup_{\delta > 0} \left(\inf_{x \in B(\bar{x}, \delta)} f(x) \right). \quad (2.46)$$

f est semi-continue inférieure en $\bar{x}$ si

$$\liminf_{x \rightarrow \bar{x}} f(x) \geq f(\bar{x}), \quad (2.47)$$

ou de manière équivalente

$$\liminf_{x \rightarrow \bar{x}} f(x) = f(\bar{x}). \quad (2.48)$$

On note que la notion de semi-continuité inférieure est plus générale que la notion de continuité.

Définition 9. L'épigraphe d'une fonction $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ est l'ensemble

$$\text{epi } f := \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid \alpha \geq f(x)\}. \quad (2.49)$$

Les courbes de niveau de f sont définies par

$$\text{lev}_{=\alpha} f := \{x \in \mathbb{R}^n \mid f(x) = \alpha\}, \quad (2.50)$$

et les courbes de niveau inférieures de f par

$$\text{lev}_{\leq \alpha} f := \{x \in \mathbb{R}^n \mid f(x) \leq \alpha\}. \quad (2.51)$$

On dit que f a des courbes de niveau inférieures bornées si pour tout $\alpha \in \mathbb{R}$, l'ensemble $\text{lev}_{\leq \alpha} f$ est borné.

Les épigraphes ont beaucoup d'utilité pour l'analyse des fonctions non lisses. Par exemple, nous verrons plus tard dans cette sous-section que pour une suite de fonctions, il est possible d'établir des propriétés de convergence intéressantes si l'on dispose d'une convergence en épigraphe. Le théorème suivant permet d'établir des liens entre les définitions vues jusqu'à présent.

Théorème 7 (73, Théorème 1.6). *Pour une fonction $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$, les propriétés suivantes sont équivalentes :*

1. f est semi-continue inférieure sur $\mathbb{R}^n$;
2. $\text{epi } f$ est un ensemble fermé dans $\mathbb{R}^n \times \mathbb{R}$;
3. les ensembles $\text{lev}_{\leq \alpha} f$ sont fermés dans $\mathbb{R}^n$ pour tout $\alpha \in \mathbb{R}$.

Enfin, le théorème suivant permet d'établir qu'il est possible d'atteindre un minimum pour f .

Théorème 8 (73, Théorème 1.9). *Soit $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ propre, semi-continue inférieure et ayant des courbes de niveau inférieures bornées. Alors la quantité $\inf f$ est finie, et l'ensemble $\text{argmin } f$ est non-vide et compact.*

Le concept qui suit permet de généraliser la notion de gradient dans le cas non lisse.

Définition 10 (73, Définition 8.3). *Soit $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ et $\bar{x} \in \mathbb{R}^n$ tel que $f(\bar{x})$ est fini. Pour $v \in \mathbb{R}^n$, on dit que*

1. v est un sous-gradient régulier de f en $\bar{x}$, écrit $v \in \hat{\partial} f(\bar{x})$, si

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x} \rangle + o(|x - \bar{x}|); \quad (2.52)$$

2. v est un sous-gradient (général) de f en $\bar{x}$, écrit $v \in \partial f(\bar{x})$, s'il existe $x_k \rightarrow \bar{x}$ avec $f(x_k) \rightarrow f(\bar{x})$ et $v_k \in \hat{\partial} f(x_k)$ avec $v_k \rightarrow v$;
3. v est un sous-gradient horizon de f en $\bar{x}$, écrit $v \in \partial^\infty f(\bar{x})$, s'il existe $x_k \rightarrow \bar{x}$ avec $f(x_k) \rightarrow f(\bar{x})$ et $v_k \in \hat{\partial} f(x_k)$ avec $\lambda_k v_k \rightarrow v$ pour $\lambda_k \searrow 0$.

Si f est continue sur un voisinage de $\bar{x}$, alors on retrouve $\partial f(\bar{x}) = \{\nabla f(\bar{x})\}$.

Il est possible d'obtenir une généralisation du théorème 1 pour le cas non lisse avec le résultat suivant.

Théorème 9 (73, Théorème 8.15). *Soit $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ propre et semi-continue inférieure sur un ensemble fermé $C \subset \mathbb{R}^n$. Soit $\bar{x} \in C$ tel que la contrainte de qualification suivante est*

satisfaite : l'ensemble $\partial^\infty f(\bar{x})$ ne contient aucun vecteur $v \neq 0$ tel que $-v \in N_C(\bar{x})$. Alors, pour que $\bar{x}$ soit un minimum local de f , la condition

$$\partial f(\bar{x}) + N_C(\bar{x}) \ni 0 \quad (2.53)$$

est nécessaire. Si f et C sont convexes, elle est suffisante, même si la contrainte de qualification n'est pas satisfaite.

Nous précisons maintenant le concept d'épi-limite inférieure et supérieure avec la définition suivante.

Définition 11 (73, Définition 7.1). *Pour une suite de fonction $\{f_k\}$, l'épi-limite inférieure $e\text{-}\lim \inf_k f_k$ est la fonction telle que*

$$\text{epi}(e\text{-}\lim \inf_k f_k) := \limsup_k (\text{epi } f_k). \quad (2.54)$$

L'épi-limite supérieure $e\text{-}\lim \sup_k f_k$ est la fonction telle que

$$\text{epi}(e\text{-}\lim \sup_k f_k) := \liminf_k (\text{epi } f_k). \quad (2.55)$$

On a $e\text{-}\lim \inf_k f_k \leq e\text{-}\lim \sup_k f_k$, et lorsque les deux coïncident, l'épi-limite $e\text{-}\lim_k f_k$ existe et est définie par

$$e\text{-}\lim_k f_k := e\text{-}\lim \inf_k f_k = e\text{-}\lim \sup_k f_k. \quad (2.56)$$

Dans ce cas, les fonctions f_k sont qualifiées d'épi-convergentes vers f , que l'on représente de la forme $f_k \xrightarrow{e} f$. Ainsi,

$$f_k \xrightarrow{e} f \iff \text{epi } f_k \longrightarrow \text{epi } f. \quad (2.57)$$

Le théorème suivant établit un résultat sur la convergence des bornes inférieures d'une suite de fonctions f_k vers la borne inférieure de l'épi-limite f , ainsi qu'une inclusion sur les ensembles des minima des f_k et des minima de f .

Théorème 10 (73, Théorème 7.33). *Si f_k a des courbes de niveaux bornées à partir d'un certain rang, que $f_k \xrightarrow{e} f$ avec f_k et f propres et semi-continues inférieures, alors*

$$\inf_x f_k(x) \rightarrow \inf_x f(x), \quad (2.58)$$

et pour tout k plus grand qu'un certain $k_0 > 0$,

$$\limsup_k (\arg \min_x f_k(x)) \subset \arg \min_x f(x). \quad (2.59)$$

2.5.2 Méthode du gradient proximal

Nous rappelons maintenant le problème (2.44) que nous voulions résoudre au début de cette section en spécifiant les hypothèses sur h :

$$\min_x f(x) + h(x), \quad (2.60)$$

où f est continûment différentiable, ∇f est Lipschitzienne de constante $L > 0$, et h est une fonction propre et semi-continue inférieure.

Pour résoudre (2.60), il n'est pas possible d'utiliser la méthode du gradient classique car h pourrait ne pas être différentiable. Si l'on effectuait un développement de Taylor d'ordre 1 avec ajout d'un terme quadratique en un point x_k de f , on pourrait chercher à trouver x_{k+1} tel que

$$\begin{aligned} x_{k+1} &\in \arg \min_x f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}\nu^{-1}\|x - x_k\|^2 + h(x) \\ &= \arg \min_x \frac{1}{2}\nu^{-1}\|x - (x_k - \nu\nabla f(x_k)^T)\|^2 + h(x). \end{aligned} \quad (2.61)$$

Le problème ci-dessus vise à trouver l'itérée suivante à l'aide d'une approximation de f , mais la contribution en h reste inchangée par rapport au problème initial. Les conditions d'optimalité du premier ordre de ce problème sont

$$0 \in x_{k+1} - x_k + \nu\nabla f(x_k) + \nu\partial h(x_{k+1}), \quad (2.62)$$

et il est alors possible de définir le *gradient généralisé*

$$G_\nu(x_k) = \nu^{-1}(x_k - x_{k+1}). \quad (2.63)$$

Si $h = 0$, on constate immédiatement que la définition du gradient généralisé correspond au gradient dans le cas lisse.

Les outils suivants seront utiles pour généraliser les méthodes d'optimisation lisse vues précédemment à des problèmes non lisses tels que (2.60). En particulier, ils vont fournir des outils pour résoudre (2.61).

Définition 12. *Pour une fonction h propre semi-continue inférieure et un paramètre $\nu > 0$,*

l'enveloppe de Moreau $e_{\nu h}$ et l'opérateur proximal $\text{prox}_{\nu h}(x)$ sont définis par

$$e_{\nu h}(x) := \inf_w \frac{1}{2} \nu^{-1} \|w - x\|^2 + h(w) = \nu^{-1} \inf_w \frac{1}{2} \|w - x\|^2 + \nu h(w), \quad (2.64a)$$

$$\text{prox}_{\nu h}(x) := \arg \min_w \frac{1}{2} \nu^{-1} \|w - x\|^2 + h(w) = \arg \min_w \frac{1}{2} \|w - x\|^2 + \nu h(w). \quad (2.64b)$$

Bien que $\text{prox}_{\nu h}$ puisse paraître difficile à calculer, il existe des solutions analytiques pour beaucoup de fonction h particulières.

Définition 13. Une fonction h propre semi-continue inférieure est dite prox-bornée s'il existe $\nu > 0$ et au moins un x tel que $e_{\nu h}(x) > -\infty$. Le seuil prox-borné ν_h est le supremum des $\nu > 0$.

La proposition suivante est une réécriture de [4, Proposition 2.3], qui est un condensé de [73, Théorème 1.25].

Proposition 2. Soit h une fonction propre semi-continue inférieure et prox-bornée avec un seuil $\nu_h > 0$. Alors, pour tout $\nu \in (0, \nu_h)$, et pour tout x ,

1. $\text{prox}_{\nu h}(x)$ est non-vide et compact,
2. $e_{\nu h}(x)$ dépend continûment en (ν, x) et $e_{\nu h}(x) \nearrow h(x)$ quand $\nu \searrow 0$.

[71] présente différents algorithmes proximaux pour l'optimisation convexe, avec des applications. La méthode du gradient proximal [44, 61] utilise l'opérateur proximal afin de générer des itérées telles que

$$x_{k+1} \in \text{prox}_{\nu h}(x_k - \nu \nabla f(x_k)), \quad (2.65)$$

pour une longueur de pas $\nu > 0$. On constate que (2.65) est exactement (2.61). En pratique, on peut arrêter l'algorithme quand la norme du gradient généralisé est suffisamment petite (nous verrons dans les chapitres suivants un choix similaire pour la condition d'arrêt). [50] montre que la convergence est linéaire sous certaines conditions.

La proposition suivante nous donne une condition pour que l'algorithme du gradient proximal soit monotone.

Proposition 3 (17, Lemme 2). Soit f continûment différentiable, ∇f Lipschitzienne de constante $L > 0$, h une fonction propre semi-continue inférieure bornée inférieurement. Pour tous $0 < \nu < 1/L$ et x_0 tel que $h(x_0) < +\infty$, x_{k+1} obtenu avec (2.65) est tel que

$$(f + h)(x_{k+1}) \leq (f + h)(x_k) - \frac{1}{2}(\nu^{-1} - L)\|x_{k+1} - x_k\|^2. \quad (2.66)$$

Un cas particulier de la méthode du gradient proximal est la méthode du gradient projeté [47, 59], où $h(x) = \chi(x \mid C)$ pour un ensemble convexe compact C . Cela nous donne un exemple de méthode pour résoudre un problème d'optimisation contraint.

2.5.3 Extensions

Il est possible d'accélérer la convergence de la méthode du gradient proximal en utilisant différentes techniques telles que celles présentées dans [12, 20, 60, 69]. Par exemple, l'algorithme FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [12], reposant sur des idées de Nesterov [66], définit

$$y_k := \text{prox}_{\nu_k h}(y_k - \nu_k \nabla f(x_k)), \quad t_{k+1} := \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \quad (2.67)$$

et met à jour

$$y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k+1}). \quad (2.68)$$

Nous ne détaillerons pas la raison du choix de t_k , mais l'idée de l'algorithme est d'améliorer la méthode du gradient proximal en exploitant une tendance sur les directions des deux itérées précédentes.

La méthode de (quasi-)Newton proximale [58] pour f et h convexes est basée sur la résolution du sous-problème

$$\arg \min_x f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k(x - x_k) + h(x), \quad (2.69)$$

et utilise l'opérateur proximal mis à l'échelle pour une matrice B définie positive

$$\text{prox}_h^B(x) := \arg \min_w \frac{1}{2}\|w - x\|_B^2 + h(w). \quad (2.70)$$

Il est possible d'utiliser $B_k = \nabla^2 f(x_k)$ si disponible (méthode de Newton proximale), ou une approximation quasi-Newton (méthode de quasi-Newton proximale). Voici une liste de références pour la résolution de (2.60) avec des méthodes quasi-Newton de recherche linéaire : [48, 57, 58, 74, 76, 77, 84]. Pour conclure cette sous-section, nous choisissons de présenter brièvement trois de ces références, dont les deux premières sont probablement parmi les plus connues, et dont la dernière est pertinente vis-à-vis de la méthode que nous allons présenter dans le chapitre 4.

PANOC [76] est un algorithme quasi-Newton proximal utilisant une recherche linéaire, implémenté pour les problèmes de contrôle optimal. Il résout des problèmes dont l'objectif a la

forme $f(x) + h(x) + g(c(x))$, où f et c sont lisses, et h et g sont non lisses.

ZeroFPR [77] est une méthode quasi-Newton proximale avec recherche linéaire non monotone pour résoudre (2.60). Elle dispose d'une convergence superlinéaire si une condition de Dennis-Moré telle que (2.21) est satisfaite.

APDNM [84] est une méthode de Newton proximale utilisant des approximations quasi-Newton diagonales définies positives pour des fonctions f et h convexes. Les résultats présentés utilisent la norme ℓ_1 pour le terme non lisse h .

2.5.4 Une méthode de région de confiance proximale utilisant des approximations quasi-Newton

Un des premiers algorithmes de région de confiance pour des problèmes d'optimisation non lisse est présenté dans [43], et traite des problèmes de la forme

$$\min_x f(x) + h(c(x)), \quad (2.71)$$

où f et c sont lisses, et h est convexe.

Dans cette sous-section, nous détaillons un algorithme de région de confiance proximal utilisant des approximations quasi-Newton, issu de [4]. Cet algorithme sera essentiel dans les chapitres suivants de cette thèse, car il est à la base des méthodes d'optimisation proximales qui seront présentées. Il peut être vu comme une adaptation de l'algorithme 2.4.1 à l'optimisation non lisse.

Nous présenterons un problème légèrement différent de (2.60) en ajoutant des contraintes de bornes :

$$\min_x f(x) + h(x) \quad \text{s.c.} \quad \ell \leq x \leq u, \quad (2.72)$$

où ℓ et u peuvent éventuellement être infinies. Ce problème plus général va être celui considéré dans les algorithmes des chapitres suivants. Bien que l'algorithme de [4] soit présenté pour (2.60), il peut être facilement adapté à (2.72) si l'on dispose d'opérateurs proximaux adaptés. Dans le cas où pour tout i , $\ell_i = -\infty$ et $u_i = +\infty$, cela revient à résoudre (2.60).

Nous définissons en particulier le modèle quadratique

$$\varphi(s; x_k, B_k) := f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B s \approx f(x_k + s), \quad (2.73a)$$

$$\psi(s; x_k) \approx h(x_k + s), \quad (2.73b)$$

$$m(s; x_k, B_k) := \varphi(s; x_k, B_k) + \psi(s; x_k), \quad (2.73c)$$

et résolvons à chaque itération le sous-problème

$$s_k \in \arg \min_s m(s; x_k, B_k) + \chi(x_k + s \mid [\ell, u] \cap (x_k + \Delta_k B)), \quad (2.74)$$

qui est un problème de région de confiance avec la contrainte supplémentaire que $x_k + s \in [\ell, u]$. On utilisera souvent $\psi(s; x_k) = h(s + x_k)$. On pourra, de manière similaire au cas lisse, calculer un point analogue au point de Cauchy, noté $s_{k,1}$, et obtenir un pas s_k tel que $m(s_k; x_k, B_k) \leq m(s_{k,1}; x_k, B_k)$. En pratique, (2.74) sera résolu avec une méthode telle que la méthode du gradient proximal en partant de $s_{k,1}$. Ce point sera calculé exactement en résolvant (2.61), mais pour être cohérent avec les notations de [4] et les prochains chapitres de cette thèse, nous allons définir

$$\varphi_{\text{cp}}(s; x_k) := f(x_k) + \nabla f(x_k)^T s, \quad (2.75a)$$

$$m(s; x_k, \nu_k) := \varphi_{\text{cp}}(s; x_k) + \frac{1}{2} \nu_k^{-1} \|s\|^2 + \psi(s; x_k), \quad (2.75b)$$

où $0 < \nu_k \leq 1/(L(x_k) + \alpha^{-1} \Delta_k^{-1})$, avec $L(x_k)$ la constante de Lipschitz de $\nabla \varphi(\cdot; x_k)$, et

$$s_{k,1} \in \arg \min_s m(s; x_k, \nu_k) + \chi(x_k + s \mid [\ell, u] \cap (x_k + \Delta_k B)). \quad (2.76)$$

Pour le critère d'arrêt de l'algorithme 2.5.1, il faut définir

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) := f(x_k) + h(x_k) - (\varphi(s_{k,1}; x_k) - \psi(s_{k,1}; x_k)). \quad (2.77)$$

Dans [4] (corrigé par [3] qui est présenté dans le chapitre 3), les auteurs montrent que $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ est un critère d'arrêt adapté. Comme $\varphi_{\text{cp}}(0; x_k) = f(x_k)$ et $\psi(0; x_k) = h(x_k)$, la définition de $s_{k,1}$ nous indique qu'on a toujours

$$\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} \geq \frac{1}{\sqrt{2}} \nu_k^{-1} \|s_{k,1}\| = \frac{1}{\sqrt{2}} \|G_{\nu_k}(x_k)\|, \quad (2.78)$$

où $G_{\nu_k}(x_k)$ est le gradient généralisé défini en (2.63). En s'inspirant de l'analyse faite dans [26], il est établi dans [4] que l'algorithme 2.5.1 met $O(\epsilon^{-2})$ itérations pour atteindre la condition d'arrêt $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} \leq \epsilon$.

Les auteurs de [7] et [31] présentent des méthodes de région de confiance similaires, avec la condition additionnelle que h soit convexe. En particulier, l'algorithme décrit dans [7] peut utiliser des évaluations inexactes de l'objectif et du gradient de f , et converge aussi en $O(\epsilon^{-2})$ itérations avec le critère d'arrêt $\nu_k^{-1} \|s_{k,1}\|$.

Algorithme 2.5.1 Méthode de région de confiance proximale quasi-Newton (TR) [4, Algorithme 3.1].

1: Choisir les constantes

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3 \leq \gamma_4, \quad \text{et} \quad \alpha > 0, \quad \beta \geq 1.$$

2: Choisir $x_0 \in \mathbb{R}^n$ où h est finie, $\Delta_0 > 0$, B_0 , calculer $f(x_0) + h(x_0)$.

3: **pour** $k = 0, 1, \dots$ **faire**

4: Choisir $0 < \nu_k \leq 1/(L(x_k) + \alpha^{-1}\Delta_k^{-1})$.

5: Définir $m(s; x_k, \nu_k)$ comme dans (2.75) et calculer $s_{k,1}$ suivant (2.76).

6: Définir $m(s; x_k, B_k)$ comme dans (2.73) et calculer s_k suivant (2.74) en remplaçant Δ_k par $\min(\Delta_k, \beta\|s_{k,1}\|)$.

7: Calculer le rapport

$$\rho_k := \frac{f(x_k) + h(x_k) - (f(x_k + s_k) + h(x_k + s_k))}{m_k(0; x_k, B_k) - m_k(s_k; x_k, B_k)}.$$

8: Si $\rho_k \geq \eta_1$, mettre à jour $x_{k+1} = x_k + s_k$, et mettre à jour B_k . Sinon, $x_{k+1} = x_k$.

9: Mettre à jour Δ_k suivant

$$\Delta_{k+1} \in \begin{cases} [\gamma_3\Delta_k, \gamma_4\Delta_k] & \text{si } \rho_k \geq \eta_2, & \text{(itération très satisfaisante)} \\ [\gamma_2\Delta_k, \Delta_k] & \text{si } \eta_1 \leq \rho_k < \eta_2, & \text{(itération satisfaisante)} \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{si } \rho_k < \eta_1 & \text{(itération non satisfaisante).} \end{cases}$$

2.5.5 Une méthode proximale de régularisation quadratique

Les auteurs de [4] présentent également un algorithme de régularisation quadratique proximal appelé R2, qui est l'analogue de l'algorithme 2.3.1 dans un contexte non lisse. Un aspect intéressant de cet algorithme est qu'il peut être utilisé pour résoudre le sous-problème (2.73) dans l'algorithme 2.5.1.

Le modèle considéré est

$$m(s; x_k, \sigma_k) := \varphi_{\text{cp}}(s; x_k) + \frac{1}{2}\sigma_k\|s\|^2 + \psi(s; x_k), \quad (2.79)$$

et à chaque itération, le sous-problème à résoudre se formule

$$s_k \in \arg \min_s m(s; x_k, \sigma_k) + \chi(x_k + s \mid [\ell, u]). \quad (2.80)$$

L'algorithme 2.5.2 décrit le fonctionnement de R2.

Algorithme 2.5.2 L'algorithme R2 [4, Algorithme 6.1].

1: Choisir les constantes

$$0 < \eta_1 \leq \eta_2 < 1, \quad \text{et} \quad 0 < \gamma_3 \leq 1 \leq \gamma_1 \leq \gamma_2.$$

2: Choisir $x_0 \in \mathbb{R}^n$, $\sigma_0 > 0$, calculer $f(x_0) + h(x_0)$.

3: **pour** $k = 0, 1, \dots$ **faire**

4: Définir $m_k(s; x_k, \sigma_k)$ comme dans (2.79) et calculer s_k suivant (2.80).

5: Calculer le rapport

$$\rho_k := \frac{f(x_k) + h(x_k) - (f(x_k + s_k) + h(x_k + s_k))}{m(0; x_k, \sigma_k) - m(s_k; x_k, \sigma_k)}.$$

6: Si $\rho_k \geq \eta_1$, $x_{k+1} = x_k + s_k$. Sinon, $x_{k+1} = x_k$.

7: Mettre à jour σ_k suivant

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

2.6 Méthodes de points intérieurs

Les problèmes linéaires de la forme

$$\min_x c^T x \quad \text{s.c.} \quad b - Ax = 0, \quad x \geq 0 \quad (2.81)$$

étaient initialement résolus avec la méthode du simplexe, découverte par George Dantzig en 1947. Si (2.81) a une solution, alors elle se trouve sur l'un des sommets du polyèdre convexe engendré par les contraintes de (2.81), et l'algorithme du simplexe consiste donc à se déplacer sur les sommets du polyèdre pour faire décroître l'objectif. Les méthodes de points intérieurs vont plutôt explorer l'intérieur du polyèdre pour trouver une solution. L'une des premières méthodes efficaces est proposée par Karmarkar [51] en 1984. Un des avantages des méthodes de points intérieurs est qu'elles peuvent être appliquées à des problèmes non linéaires et non convexes, comme nous allons le voir dans cette section.

2.6.1 Problèmes barrières pour les contraintes d'inégalités

Nous nous intéressons maintenant au problème

$$\min_x f(x) \quad \text{s.c.} \quad x \geq 0, \quad (2.82)$$

dont les conditions de KKT sont

$$\nabla f(x) - z = 0, \quad (2.83a)$$

$$x \geq 0, \quad (2.83b)$$

$$z \geq 0, \quad (2.83c)$$

$$x_i z_i = 0. \quad (2.83d)$$

Ce problème peut être remplacé par le problème barrière

$$\min_x f(x) - \mu \sum_i \log x_i, \quad (2.84)$$

ayant pour condition d'optimalité du premier ordre

$$\nabla f(x) - \mu X^{-1}e = 0, \quad x > 0. \quad (2.85)$$

Le terme de barrière $\mu \sum_i \log x_i$ a pour but d'éloigner x de zéro afin de garder un point réalisable pour (2.82). Pour un algorithme utilisant une suite $\mu_k \searrow 0$, on espère qu'à mesure que μ_k s'approche de 0, les solutions des sous-problèmes (2.84) avec $\mu = \mu_k$ s'approchent de la solution de (2.82).

Si on pose $f_k(x) := f(x) - \mu_k \sum_i \log x_i$, il est possible de montrer que $f_k \xrightarrow{e} \tilde{f}$, où $\tilde{f} := f(x) + \chi(x; \mathbb{R}_+^n)$ (voir [6]), et que les courbes de niveau de f sont bornées, permettant d'appliquer le théorème 10. [81] présente différents résultats théoriques pour les méthodes de points intérieurs dans différents contextes (programmation linéaire, non linéaire, non convexe), et établit notamment la convergence locale des méthodes barrières pour des problèmes tels que (2.82).

Pour l'optimisation non lisse, [37] et [32] utilisent la méthode du gradient proximal pour résoudre les sous-problèmes barrière (2.84) généralisés au cas non lisse.

2.6.2 Méthode de région de confiance pour les problèmes barrières

[34, Chapitre 13.6] décrit une méthode pour résoudre (2.82). L'idée est d'utiliser un algorithme de points intérieurs où chaque problème barrière sera résolu par une méthode de région de confiance.

Considérons un modèle de Taylor d'ordre 2 du problème barrière à l'itération k pour un point

$x > 0$

$$m_k(x + s) = f(x) - \mu_k \sum_i \log x_i + \nabla f(x)^T s - \mu_k x^{-T} s + \frac{1}{2} s^T (\nabla^2 f(x) + \mu_k X^{-2}) s. \quad (2.86)$$

Un inconvénient de ce modèle est que son Hessien $(\nabla^2 f(x) + \mu_k X^{-2})$ est mal conditionné lorsqu'une composante x_i s'approche de zéro trop rapidement. Les méthodes primales-duales utilisent plutôt un Hessien de la forme $(\nabla^2 f(x) + X^{-1}Z)$, où $z > 0$. Si $z = \mu_k X^{-1}e$, alors on retrouve le modèle (2.86).

L'algorithme 2.6.1 est une méthode de région de confiance pour la résolution du problème barrière (2.84) à l'itération k . Afin de rendre (2.83d) plus facile à résoudre, on utilise une perturbation μ_k qui mène à l'équation $x_i z_i = \mu_k$. En appliquant la méthode de Newton, on obtient alors

$$X\Delta z + Z\Delta x = \mu_k e - XZe, \quad (2.87)$$

d'où l'expression de $\bar{z}_{k,j+1}$ dans l'algorithme 2.6.1, obtenue en remplaçant Δx dans (2.87) par $s_{k,j}$. La projection sur l'intervalle de (2.90) permet de s'assurer que $z_{k,j+1} > 0$, ainsi que de vérifier des conditions utiles pour montrer les propriétés de convergence de la méthode (voir [34, Chapitre 13.6] pour plus de détails).

Algorithme 2.6.1 Méthode de région de confiance pour les itérations internes [34, Algorithme 13.6.1].

1: Choisir les constantes

$$\epsilon_k > 0, \quad 0 < \eta_1 \leq \eta_2 < 1, \quad 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3 \leq \gamma_4, \quad \text{et} \quad \alpha > 0, \quad \beta \geq 1.$$

2: Choisir $x_{k,0} \in \mathbb{R}^n$ où h est finie, $\Delta_{k,0} > 0$, calculer $f(x_{k,0})$.

3: **pour** $j = 0, 1, \dots$ **faire**

4: Définir un modèle $m_{k,j}$ de la forme

$$m_{k,j}(x_{k,j} + s) = m_{k,j}^f(x_{k,j} + s) - \mu_k \sum_i \log((x_{k,j})_i) - \mu_k x_{k,j}^{-T} s + \frac{1}{2} s^T X_{k,j}^{-1} Z_{k,j} s \quad (2.88)$$

où $m_{k,j}^f$ est un modèle de f vérifiant certaines hypothèses (non mentionnées ici, mais explicitées dans [34, Algorithme 3.1])

5: Calculer un pas $s_{k,j} \in \Delta_{k,j} \mathbb{B}$ tel que $\text{dist}(x_{k,j} + s_{k,j}, \partial \mathbb{R}_+^n) \geq \epsilon_k \text{dist}(x_{k,j}, \partial \mathbb{R}_+^n)$ et tel que s_k diminue suffisamment le modèle $m_{k,j}$ (voir [34, Chapitre 13.6] pour plus de détails).

6: Calculer le rapport

$$\rho_{k,j} := \frac{f(x_{k,j}) - \mu_k \sum_i \log((x_{k,j})_i) - (f(x_{k,j} + s_{k,j}) - \mu_k \sum_i \log((x_{k,j} + s_{k,j})_i))}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})}.$$

7: Si $\rho_{k,j} \geq \eta_1$, mettre à jour $x_{k,j+1} = x_{k,j} + s_{k,j}$. Sinon, $x_{k,j+1} = x_{k,j}$.

8: Mettre à jour $\Delta_{k,j}$ suivant

$$\Delta_{k,j+1} \in \begin{cases} [\gamma_3 \Delta_{k,j}, \gamma_4 \Delta_{k,j}] & \text{si } \rho_{k,j} \geq \eta_2, & \text{(itération très satisfaisante)} \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{si } \eta_1 \leq \rho_{k,j} < \eta_2, & \text{(itération satisfaisante)} \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{si } \rho_{k,j} < \eta_1 & \text{(itération non satisfaisante).} \end{cases}$$

9: Calculer

$$\bar{z}_{k,j+1} = \mu_k X_{k,j}^{-1} e - X_{k,j}^{-1} Z_{k,j} s_{k,j}, \quad (2.89)$$

et en déduire $z_{k,j+1}$ en projetant $\bar{z}_{k,j+1}$ composante par composante sur l'intervalle

$$[\kappa_{\text{zul}} \min(e, z_{k,j}, \mu_k X_{k,j+1}^{-1} e), \max(\kappa_{\text{zuu}} e, z_{k,j}, \kappa_{\text{zuu}} \mu_k^{-1} e, \kappa_{\text{zuu}} \mu_k X_{k,j+1}^{-1} e)], \quad (2.90)$$

où $0 < \kappa_{\text{zul}} < 1 < \kappa_{\text{zuu}}$.

Dans l'algorithme 2.6.2, (2.91a) permet de s'assurer de la vérification de (2.83a), (2.91b) permet de s'assurer de la proximité au chemin central. La condition suffisante du deuxième ordre donnée par le théorème 4 donne lieu à (2.91c).

Algorithme 2.6.2 Méthode de région de confiance pour les itérations externes [34, Algorithme 13.6.2].

- 1: Choisir $x_0 \in \mathbb{R}_+^n$, $z_0 \in \mathbb{R}_+^n$ et $\mu_0 > 0$.
- 2: **pour** $j = 0, 1, \dots$ **faire**
- 3: Choisir $\epsilon_k \in (0, 1)$.
- 4: Minimiser $f(x) - \mu_k \sum_i \log(x)$ en utilisant l'algorithme 2.6.1.
- 5: Arrêter l'algorithme 2.6.1 dès que $(x_{k,j}, z_{k,j}) = (x_{k+1}, z_{k+1})$ vérifie

$$\|\nabla f(x_{k+1}) - z_{k+1}\| \leq \epsilon_k, \quad (2.91a)$$

$$\|X_{k+1}Z_{k+1} - \mu_k I\| \leq \epsilon_k, \quad (2.91b)$$

$$\lambda_{\min}(\nabla^2 f(x_{k+1}) + X_{k+1}^{-1}Z_{k+1}) \geq -\epsilon_k, \quad (2.91c)$$

avec $x_{k+1} > 0$ et $z_{k+1} > 0$.

- 6: Choisir μ_{k+1} de telle sorte que $\lim_{k \rightarrow +\infty} \mu_k = 0$.
-

Parmi les algorithmes de région de confiance pour les problèmes barrières utilisables sur des problèmes légèrement plus généraux que (2.82), nous pouvons également citer [21, 22, 33, 85].

CHAPITRE 3 CORRIGENDUM: A PROXIMAL QUASI-NEWTON TRUST-REGION METHOD FOR NONSMOOTH REGULARIZED OPTIMIZATION

Afin de compléter la revue de littérature, ce chapitre inclut des corrections à l'article [4]. Ces corrections portent principalement sur la mesure de criticité utilisée pour arrêter l'algorithme 2.5.1 (TR). Dans le cas où $h = 0$, la mesure est maintenant équivalente à contrôler $\|\nabla f(x_k)\|$, qui est la mesure généralement utilisée pour résoudre des problèmes lisses. Nous introduisons ξ_{cp} afin de mettre en lumière certains points communs avec le point de Cauchy dans le cas lisse présenté dans la section 2.4.2, et notamment l'usage d'un modèle basé sur une approximation de Taylor d'ordre 1. Enfin, nous justifions que les résultats présentés dans [4] continuent de s'appliquer avec ces modifications.

Title: Corrigendum: a proximal quasi-Newton trust-region method for nonsmooth regularized optimization

Authors: Aleksandr Aravkin, Robert Baraldi, Geoffroy Leconte and Dominique Orban

Abstract: The purpose of the present note is to bring clarifications to certain concepts and surrounding notation of [4]. All results therein continue to hold. The clarifications ensure complete agreement between the paper and our implementations.

In (3.2), we should have defined $\varphi(s; x) := \varphi_{\text{cp}}(s; x) + \frac{1}{2}\nu^{-1}\|s\|^2$ where φ_{cp} is the first order Taylor approximation

$$\varphi_{\text{cp}}(s; x) = f(x) + \nabla f(x)^T s, \quad (3.1)$$

and “cp” stands for “Cauchy point”. In (3.4), $s_{k,1}$ could have been defined equivalently as the solution of

$$s_{k,1} \in \underset{s}{\operatorname{argmin}} \quad \varphi_{\text{cp}}(s; x_k) + \frac{1}{2}\nu^{-1}\|s\|^2 + \psi(s; x_k) + \chi(s; \Delta_k). \quad (3.2)$$

The first step of the proximal-gradient algorithm applied to minimize $\varphi_{\text{cp}}(s; x_k) + \psi(s; x_k) + \chi(s; \Delta_k)$ with step size ν is then precisely $s_{k,1}$.

In addition to the definition of ξ in (3.6), we also define

$$\xi_{\text{cp}}(\Delta; x_k, \nu) := f(x_k) + h(x_k) - \varphi_{\text{cp}}(s_{k,1}; x_k) - \psi(s_{k,1}; x_k). \quad (3.3)$$

Note that for $\varphi = \varphi_{\text{cp}}$,

$$\begin{aligned}\xi(\Delta_k; x_k, \nu_k) &= (f + h)(x_k) - (\varphi_{\text{cp}}(s_{k,1}; x_k) + \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2 + \psi(s_{k,1}; x_k)) \\ &\leq (f + h)(x_k) - (\varphi_{\text{cp}}(s_{k,1}; x_k) + \psi(s_{k,1}; x_k)) \\ &= \xi_{\text{cp}}(\Delta_k; x_k, \nu_k).\end{aligned}$$

Thus $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = 0 \implies \xi(\Delta_k; x_k, \nu_k) = 0$, and, by Proposition 3.3, x_k is first-order stationary. Using the definition of $s_{k,1}$, we have

$$\varphi_{\text{cp}}(s_{k,1}; x_k) + \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2 + \psi(s_{k,1}; x_k) \leq \varphi_{\text{cp}}(0; x_k) + \psi(0; x_k).$$

The above inequality, combined with the facts that $\varphi(0; x_k) = f(x_k)$ and $\psi(0; x_k) = h(x_k)$, leads to

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = f(x_k) + h(x_k) - (\varphi_{\text{cp}}(s_{k,1}; x_k) + \psi(s_{k,1}; x_k)) \geq \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2. \quad (3.4)$$

In Step Assumption 3.1, (3.8b) should read

$$m_k(0; x_k) - m_k(s_k; x_k) \geq \kappa_{\text{mdc}}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k). \quad (3.5)$$

The justification of (3.8b) (below Model Assumption 3.2) is no longer relevant with the above change. However, the following proposition shows that (3.5) in Step Assumption 3.1 can still be satisfied if

$$\varphi(s; x_k, B_k) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2}s^T B_k s = \varphi_{\text{cp}}(s; x_k) + \frac{1}{2}s^T B_k s, \quad (3.6)$$

where $\{B_k\}$ is bounded. In addition, we require the existence of a maximal trust-region radius $\Delta_{\text{max}} \geq \Delta_0$. This can be enforced in Algorithm 3.1 at the end of Step 11 with the following assignment:

$$\Delta_{k+1} \leftarrow \min(\Delta_{k+1}, \Delta_{\text{max}}). \quad (3.7)$$

Proposition 4. *Let Algorithm 3.1 be modified according to (3.7), and $\varphi(s; x_k, B_k)$ defined in (3.6) be used in Step 5. Let Model Assumption 3.2 be satisfied, and the sequence of Hessian approximations $\{B_k\}$ generated by Algorithm 3.1 be bounded. Then, there exists $\kappa_{\text{mdc}} \in (0, 1)$ such that (3.5) holds for all k .*

Proof. By definition of s_k , computed in Step 8 of Algorithm 3.1 using (3.6) in Step 5,

$$\begin{aligned}\varphi(s_k; x_k, B_k) + \psi(s_k; x_k) &= m(s_k; x_k, B_k) \leq m(s_{k,1}; x_k, B_k) \\ &= \varphi_{\text{cp}}(s_{k,1}; x_k) + \frac{1}{2}s_{k,1}^T B_k s_{k,1} + \psi(s_{k,1}; x_k) \\ &\leq \varphi_{\text{cp}}(s_{k,1}; x_k) + \frac{1}{2}\|B_k\|\|s_{k,1}\|^2 + \psi(s_{k,1}; x_k).\end{aligned}$$

We multiply the above inequality by -1 and add $m(0; x_k, B_k) = m(0; x_k, \nu_k) = f(x_k) + h(x_k)$ to both sides to obtain

$$m(0; x_k, B_k) - m(s_k; x_k, B_k) \geq \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) - \frac{1}{2}\|B_k\|\|s_{k,1}\|^2.$$

To satisfy (3.5), it is sufficient to show that there exists $\kappa_{\text{mdc}} \in (0, 1)$ such that

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) - \frac{1}{2}\|B_k\|\|s_{k,1}\|^2 \geq \kappa_{\text{mdc}}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k),$$

i.e.,

$$(1 - \kappa_{\text{mdc}})\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2}\|B_k\|\|s_{k,1}\|^2.$$

Using (3.4), it is also sufficient to show that there exists $\kappa_{\text{mdc}} \in (0, 1)$ such that

$$(1 - \kappa_{\text{mdc}})\nu_k^{-1} \geq \|B_k\|. \quad (3.8)$$

If $\|B_k\| = 0$, (5.21) is satisfied for any $\kappa_{\text{mdc}} \in (0, 1)$. Otherwise, because (3.6) indicates that the Lipschitz constant $L(x_k)$ of $\nabla\varphi(\cdot; x_k, B_k)$ in Model Assumption 3.2 is $L(x_k) = \|B_k\|$, the definition of ν_k in Step 4 of Algorithm 3.1 yields

$$\|B_k\|\nu_k = \frac{\|B_k\|}{\|B_k\| + \alpha^{-1}\Delta_k^{-1}} \leq \frac{1}{1 + \alpha^{-1}\Delta_k^{-1}\|B_k\|^{-1}} \leq \frac{1}{\alpha^{-1}\Delta_{\max}^{-1}(\sup_{k \in \mathbb{N}}\|B_k\|)^{-1} + 1} \in (0, 1), \quad (3.9)$$

because $\{B_k\}$ is bounded by assumption. We deduce from (5.22) that (5.21), and therefore (3.5) hold. $\square$

In the proof of Theorem 3.4, (3.4) guarantees that the inequality

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2 \geq \frac{1}{2}(L(x_k) + \alpha^{-1}\Delta_k^{-1})\|s_{k,1}\|^2 \geq \frac{1}{2}\alpha^{-1}\Delta_k^{-1}\|s_{k,1}\|^2, \quad (3.10)$$

continues to hold, where $L(x_k)$ is the Lipschitz constant of the gradient of $\varphi(\cdot; x_k)$. Thus, the theorem remains valid.

In the simple case where $h = 0$ and $s_{k,1} = -\nu_k \nabla f(x_k)$ is inside the trust-region,

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = -\nabla f(x_k)^T s_{k,1} = \nu_k \|\nabla f(x_k)\|^2, \quad (3.11)$$

which suggests that the appropriate criticality measure (analogous to the gradient norm in the smooth case) should be $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ instead of $\nu_k^{-1} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$. The first-order optimality condition in (3.13) then becomes

$$\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_{\min}; x_k, \nu_k)^{1/2} \leq \epsilon \quad (0 < \epsilon < 1). \quad (3.12)$$

This changes the bound in (3.15) to

$$|S(\epsilon)| \leq \frac{(f+h)(x_0) - (f+h)_{\text{low}}}{\eta_1 \kappa_{\text{mdc}} \nu_{\min} \epsilon^2} = O(\epsilon^{-2}), \quad (3.13)$$

i.e., $\nu_{\min}^2$ becomes $\nu_{\min}$, but does not alter the order of complexity.

Similarly, the limits considered in Corollary 3.9 and Theorem 3.11 should be

$$\liminf_{k \rightarrow \infty} \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$$

and

$$\lim_{k \rightarrow \infty} \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2},$$

respectively.

In addition to the definition of $\xi(\sigma; x)$ in (6.4), we should have also defined

$$\xi_{\text{cp}}(\sigma; x) := f(x) + h(x) - (\varphi(s; x) + \psi(s; x)), \quad (3.14)$$

where

$$s \in \arg \min_s m(s; x, \sigma). \quad (3.15)$$

However, (6.6) is still valid.

After (6.10), by analogy with Algorithm 3.1, the stationarity measure should be

$$\sigma_k^{1/2} \xi_{\text{cp}}(\sigma_{\max}; x_k)^{1/2}.$$

In Section 7, all our numerical experiments use $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ and $\xi(\sigma_k; x_k)^{1/2}$ as proxies for the stationarity measures. Even though the factors $\nu_k^{-1/2}$ and $\sigma_k^{1/2}$, respectively, are not included in the proxies, the measures remain valid as long as $\nu_{\min} > 0$ and $\sigma_{\max} < \infty$, which

is the case under our assumptions.

CHAPITRE 4 ARTICLE 1: THE INDEFINITE PROXIMAL GRADIENT METHOD

Nous nous plaçons dans le cadre de la résolution de (2.72) avec h étant une fonction séparable, c'est-à-dire que pour tout $x \in \mathbb{R}^n$, $h(x) = \sum_{i=1}^n h_i(x_i)$, où $h_i : \mathbb{R} \mapsto \mathbb{R} \cup \{+\infty\}$. Considérons le modèle m défini en (2.73), et remplaçons B_k par une approximation quasi-Newton diagonale D_k . Dans ce cas, pour certains régulariseurs h tels que la norme ℓ_1 et la norme ℓ_0 (qui sont séparables), nous allons expliciter une solution analytique pour la résolution du sous-problème (2.74) dans l'algorithme 2.5.1. Il existe plusieurs techniques similaires dans la littérature (mentionnées plus tard dans ce chapitre), consistant à mettre à l'échelle les pas de la méthode du gradient proximal avec D_k . Cependant, ces méthodes fonctionnent uniquement dans le cas où D_k est définie positive. L'algorithme présenté dans ce chapitre, nommé TRDH (*Trust-Region method with Diagonal Hessian approximations*) ne souffre pas de cette restriction, car il peut traiter des matrices D_k indéfinies. Dans le cas où il serait souhaitable de conserver une approximation quasi-Newton non diagonale B_k , TRDH est aussi utilisable comme sous-solveur pour TR (algorithme 2.5.1).

À chaque itération, TRDH effectue une évaluation d'opérateur proximal nécessaire au calcul de la mesure de criticité $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k, x_k, \nu_k)^{1/2}$ identique à celle utilisée pour TR dans le chapitre 3, et une résolution analytique du sous-problème de région de confiance. Nous montrons des propriétés de convergence similaires à celles de TR, et notamment une borne de complexité de ϵ^{-2} .

Nous présentons également une variante basée sur un critère d'arrêt légèrement différent, qui permet d'économiser l'évaluation de l'opérateur proximal, et donc de résoudre uniquement le sous-problème de région de confiance.

Nous effectuons des comparaisons sur plusieurs problèmes entre TRDH, R2 (algorithme 2.5.2), et TR (algorithme 2.5.1) avec R2 et TRDH comme sous-solveurs, en testant plusieurs approximations quasi-Newton différentes, et nous observons qu'une des variantes de TRDH ou de TR avec TRDH comme sous-solveur est toujours meilleure que R2 ou TR avec R2 comme sous-solveur.

Title: The indefinite proximal gradient method

Authors: Geoffroy Leconte and Dominique Orban

Submitted to: Computational Optimization and Applications

Submission date: 15/11/2023

Abstract: We introduce a variant of the proximal gradient method in which the quadratic term is diagonal but may be indefinite, and is safeguarded by a trust region. Our method is a special case of the proximal quasi-Newton trust-region method of Aravkin *et al.* [4]. We provide closed-form solution of the step computation in certain cases where the nonsmooth term is separable and the trust region is defined in the infinity norm, so that no iterative subproblem solver is required. Our analysis expands upon that of [4] by generalizing the trust-region approach to problems with bound constraints. We provide an efficient open-source implementation of our method, named TRDH, in the Julia language in which Hessians approximations are given by diagonal quasi-Newton updates. TRDH evaluates one standard proximal operator and one indefinite proximal operator per iteration. We also analyze and implement a variant named iTRDH that performs a single indefinite proximal operator evaluation per iteration. We establish that iTRDH enjoys the same asymptotic worst-case iteration complexity as TRDH. We report numerical experience on unconstrained and bound-constrained problems, where TRDH and iTRDH are used both as standalone and subproblem solvers. Our results illustrate that, as standalone solvers, TRDH and iTRDH improve upon the quadratic regularization method R2 of [4] but also sometimes upon their quasi-Newton trust-region method, referred to here as TR-R2, in terms of smooth objective value and gradient evaluations. On challenging nonnegative matrix factorization, binary classification and data fitting problems, TRDH and iTRDH used as subproblem solvers inside TR improve upon TR-R2 for at least one choice of diagonal approximation.

In honor and memory of our colleague and friend Daniela di Serafino.

4.1 Introduction

We consider the nonsmooth regularized problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + h(x) \quad \text{subject to } \ell \leq x \leq u, \quad (4.1)$$

where $\ell \in (\mathbb{R} \cup \{-\infty\})^n$, $u \in (\mathbb{R} \cup \{+\infty\})^n$ with $\ell \leq u$ componentwise, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable on an open set containing $[\ell, u]$, and $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper and lower semicontinuous (lsc). A component $\ell_i = -\infty$ or $u_i = +\infty$ indicates that x_i is unbounded below or above, respectively. Both f and h may be nonconvex. Typically, h is nonsmooth and serves to identify a local minimizer of f with desirable features, such as sparsity.

Numerical methods for (6.1) are typically based on the proximal-gradient method [61]. Aravkin *et al.* [4] provide an overview of recent works focusing on (6.1) where both f and h may be nonconvex, to which we refer the reader. In addition, they propose two methods: an adaptive quadratic regularization approach named R2, which may be viewed as a proximal-gradient method with adaptive step size, and a quasi-Newton trust-region method named TR in which the subproblem consist in minimizing a quadratic approximation of f about the current iterate regularized by a model of h inside a trust region. Typically, an explicit solution to the subproblem is not known, and an iterative procedure must be used—the authors use R2. Aravkin *et al.* [2] develop a similar approach designed for applications where f is a least-squares residual. They propose a trust-region and a regularization approach. Again, they use R2 as subproblem solver.

In both R2 and the traditional proximal-gradient method, a uniformly positively scaled ∇f is used to compute a step, and that computation relies on the proximal operator—see Section 6.2 for precise definitions. Some authors consider positive-definite diagonal gradient scaling—see below. In the present research, we contend that generalizing the definition of the proximal operator by allowing a diagonal scaling of ∇f , and even permitting negative scaling factors, continues to allow us to derive analytical solutions for several nonsmooth terms of interest in applications. Moreover, such potentially indefinite scaling might allow the model to better capture inherent nonconvexity in f about the current iterate. We devise a trust-region method based on those ideas and name it TRDH, which stands for *trust-region method with diagonal Hessian approximations*. At every iteration, our method performs the evaluation of both a classical proximal operator and a generalized proximal operator with indefinite diagonal scaling. However, it is possible to modify TRDH slightly to devise a variant that requires a single generalized proximal operator evaluation per iteration while preserving the asymptotic worst-case evaluation complexity bound. We name the variant iTRDH, which stands for *indefinite trust-region method with diagonal Hessian*.

Diagonal gradient scaling could be referred to as diagonal quasi-Newton, and though the literature appears to be thin on that subject, there exist a few references. Diagonal Hessian approximations range from a multiple of the identity, as in the traditional proximal-gradient or the spectral gradient method, to approximations computed based on a relaxed secant equation. We go into more details in Section 4.4.

Because we use the ℓ_∞ -norm to define the trust region, we are able to treat bound constraints naturally, by taking the intersection of $[\ell, u]$ with the trust region. Our generalized *indefinite* proximal operators must take the indicator of the resulting box into account together with h , or a model thereof.

A by-product of the present research is an efficient software implementation of TRDH and iTRDH, both as standalone solvers, and as subproblem solvers for use inside TR. TRDH and iTRDH can use several diagonal Hessian approximations: the multiple of the identity given by the spectral gradient approximation, and two diagonal quasi-Newton approximations based on the weak secant equation. Our open-source implementations in Julia are available from <https://github.com/JuliaSmoothOptimizers/RegularizedOptimization.jl>.

We report numerical experience conducted with TRDH, iTRDH, TR-TRDH and TR-iTRDH on a range of unconstrained and bound-constrained problems, where TR-TRDH and TR-iTRDH refer to TRDH and iTRDH used as subsolvers in TR. Our general conclusion is that for at least one choice of a diagonal Hessian approximation, TRDH and iTRDH outperform R2 in terms of evaluations of f and ∇f . In addition, for at least one choice of a diagonal Hessian approximation, TR-TRDH and TR-iTRDH outperform TR-R2 on the same metric. Our detailed results are in Section 6.6.

Related research

Most of the literature focuses on positive-definite diagonal scaling of the proximal operator; Becker and Fadili [13] and Becker *et al.* [14] consider positive-definite quasi-Newton approximations of the form $H = D + VV^T$ where D is positive definite and diagonal, and a specialized procedure to solve the proximal quasi-Newton subproblems. Under their assumptions, H remains uniformly bounded. Their analysis is restricted to f and h convex and does not provide complexity bounds.

Duchi *et al.* [40] present the ADAGRAD algorithm, which is commonly used in online learning. ADAGRAD is a stochastic algorithm but a deterministic implementation of it would minimize a smooth objective such as f using a variant of the projected gradient algorithm, that can be seen as a special case of the proximal gradient method, using a positive diagonal gradient scaling. At iteration k , the scaling is set to $\text{diag}(G_k)^{-1/2}$ where $G_k = \sum_{j=0}^k \nabla f(x_j) \nabla f(x_j)^T$.

Scheinberg and Tang [74] also focus on positive-definite and uniformly-bounded Hessian approximations for f and h convex. Their numerical results employ a limited-memory BFGS approximation, although the latter is liable to grow unbounded [34, §8.4]. The proximal quasi-Newton subproblem is solved inexactly with a coordinate descent algorithm.

Aravkin *et al.* [4] propose a proximal quasi-Newton trust-region method for nonconvex f and h under weak assumptions, accompanied by a complexity analysis. They employ limited-memory BFGS or SR1 approximations and solve subproblems using the proximal-gradient method or R2. Though their analysis assumes the Hessian approximations are uniformly bounded, there

are known procedures to estimate bounds on the eigenvalues of quasi-Newton approximations after an update, and those bounds can be used to skip or modify the update to keep it bounded [62].

Similarly, Baraldi and Kouri [7] present a proximal trust-region method with inexact objective and gradient evaluations, under the additional assumption that h must be convex. They show numerical results with exact Hessian when it is available, and employ iterative solvers to approximate it otherwise.

Notation

$\mathbb{B}$ is the unit ball centered at the origin and defined by a norm dictated by the context, and $\Delta\mathbb{B}$ is the ball of radius $\Delta > 0$ centered at the origin. For fixed $x \in \mathbb{R}^n$, the ball of radius Δ centered at x is $x + \Delta\mathbb{B}$. When it is necessary to indicate that $\mathbb{B}$ is defined by the ℓ_p -norm, for $1 \leq p \leq +\infty$, we write $\mathbb{B}_p$. For $A \subseteq \mathbb{R}^n$, the indicator of A is $\chi(\cdot \mid A) : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ defined as $\chi(x \mid A) = 0$ if $x \in A$ and $+\infty$ otherwise. If $A \neq \emptyset$, $\chi(\cdot \mid A)$ is proper. If A is closed, $\chi(\cdot \mid A)$ is lsc. For $D \in \mathbb{R}^{m \times n}$ with elements $\delta_{i,j}$, $|D| \in \mathbb{R}^{m \times n}$ has elements $|\delta_{i,j}|$. For a finite set $A \subset \mathbb{N}$, we denote $|A|$ its cardinality. If f_1 and f_2 are two positive functions of $\epsilon > 0$, we say that $f_1(\epsilon) = O(f_2(\epsilon))$ if there exists a constant $C > 0$ such that $f_1(\epsilon) \leq C f_2(\epsilon)$ for all $\epsilon > 0$ sufficiently small.

4.2 Background

The following are standard variational analysis concepts—see, e.g., [73]. Let $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $\bar{x} \in \mathbb{R}^n$ where ϕ is finite. The Fréchet subdifferential of ϕ at $\bar{x}$ is the closed convex set $\hat{\partial}\phi(\bar{x})$ of $v \in \mathbb{R}^n$ such that

$$\liminf_{\substack{x \rightarrow \bar{x} \\ x \neq \bar{x}}} \frac{\phi(x) - \phi(\bar{x}) - v^T(x - \bar{x})}{\|x - \bar{x}\|} \geq 0.$$

The limiting subdifferential of ϕ at $\bar{x}$ is the closed, but not necessarily convex, set $\partial\phi(\bar{x})$ of $v \in \mathbb{R}^n$ for which there exist $\{x_k\} \rightarrow \bar{x}$ and $\{v_k\} \rightarrow v$ such that $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$ and $v_k \in \hat{\partial}\phi(x_k)$ for all k . $\hat{\partial}\phi(\bar{x}) \subset \partial\phi(\bar{x})$ always holds.

The horizon subdifferential of ϕ at $\bar{x}$ is the closed, but not necessarily convex, cone $\partial^\infty\phi(\bar{x})$ of $v \in \mathbb{R}^n$ for which there exist $\{x_k\} \rightarrow \bar{x}$, $\{v_k\}$ and $\{\lambda_k\} \downarrow 0$ such that $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$, $v_k \in \hat{\partial}\phi(x_k)$ for all k , and $\{\lambda_k v_k\} \rightarrow v$.

If $C \subseteq \mathbb{R}^n$ and $\bar{x} \in C$, the closed convex cone $\widehat{N}_C(\bar{x}) := \hat{\partial}\chi(\bar{x} \mid C)$ is called the regular normal cone to C at $\bar{x}$. The closed cone $N_C(\bar{x}) := \partial\chi(\bar{x} \mid C) = \partial^\infty\chi(\bar{x} \mid C)$ is called the normal cone to C at $\bar{x}$. $\widehat{N}_C(\bar{x}) \subseteq N_C(\bar{x})$ always holds, and is an equality if C is convex.

If $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ is proper lsc, and $C \subseteq \mathbb{R}^n$ is closed, we say that the *constraint qualification* is satisfied at $\bar{x} \in C$ for the constrained problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \phi(x) \quad \text{subject to } x \in C$$

if

$$\partial^\infty \phi(\bar{x}) \cap N_C(\bar{x}) = \{0\}. \quad (4.2)$$

As an example where (5.7) fails to hold, let $n = 1$, $f(x) = 0$ for all $x \in \mathbb{R}$,

$$h(x) = \begin{cases} x & \text{if } x \geq 0, \\ +\infty & \text{if } x < 0, \end{cases}$$

and $C = [0, 1]$. Clearly, $\bar{x} = 0$ is the only stationary point and $N_C(\bar{x}) = \{v \mid v \leq 0\}$. Let $\text{epi } h$ denote the epigraph of h , i.e., the set $\{(x, t) \mid x \in \mathbb{R}^n, t \geq h(x)\}$. Because h is locally lsc about $\bar{x}$, [73, Theorem 8.9] yields

$$\partial^\infty h(\bar{x}) = \{v \mid (v, 0) \in N_{\text{epi } h}(\bar{x}, h(\bar{x}))\} = \{v \mid (v, 0) \in N_{\text{epi } h}(0, 0)\} = N_C(\bar{x}),$$

and the constraint qualification does not hold. The reason is that h and $\chi(\cdot \mid C)$ both have a jump discontinuity at $\bar{x}$ and take the value $+\infty$ for $x < \bar{x}$, which is akin to repeating a constraint.

We say that $\bar{x}$ is first-order stationary for (6.1) if $0 \in \partial(f + h + \chi(\cdot \mid [\ell, u]))(\bar{x}) = \nabla f(\bar{x}) + \partial(h + \chi(\cdot \mid [\ell, u]))(\bar{x})$. If $\bar{x}$ is a local solution of (6.1), it is first-order stationary [73, Theorem 10.1]. Under (5.7) applied to (6.1), which reads $(\nabla f(\bar{x}) + \partial^\infty h(\bar{x})) \cap N_{[\ell, u]}(\bar{x}) = \{0\}$, the necessary optimality condition can be written equivalently as $0 \in \nabla f(\bar{x}) + \partial h(\bar{x}) + N_{[\ell, u]}(\bar{x})$ [73, Theorem 8.15]. When (6.1) is unconstrained or $\bar{x} \in \text{int}[\ell, u]$, the normal cone is $\{0\}$, the constraint qualification is satisfied, and the necessary condition reduces to $0 \in \nabla f(\bar{x}) + \partial h(\bar{x})$.

The proximal operator associated with h is

$$\text{prox}_{\nu h}(q) := \underset{x}{\text{argmin}} \quad \frac{1}{2}\nu^{-1}\|x - q\|_2^2 + h(x), \quad (4.3)$$

where $\nu > 0$ is a preset steplength. If h is prox-bounded and $\nu > 0$ is sufficiently small, $\text{prox}_{\nu h}(q)$ is a nonempty and closed set. It may contain multiple elements.

The proximal gradient method [61] is a generalization of the gradient method that takes the

nonsmooth term into account. It generates iterates $\{x_k\}$ according to

$$x_{k+1} \in \underset{\nu_k h}{\text{prox}}(x_k - \nu_k \nabla f(x_k)). \quad (4.4)$$

Equivalently, and more instructively,

$$s_k \in \underset{s}{\text{argmin}} \quad \frac{1}{2} \nu_k^{-1} \|s + \nu_k \nabla f(x_k)\|_2^2 + h(x_k + s), \quad (4.5a)$$

$$= \underset{s}{\text{argmin}} \quad \nabla f(x_k)^T s + \frac{1}{2} \nu_k^{-1} \|s\|_2^2 + h(x_k + s), \quad (4.5b)$$

$$x_{k+1} = x_k + s_k, \quad (4.5c)$$

where it becomes clear that the computation of s_k results from formulating a simple quadratic model

$$f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} \nu_k^{-1} \|s\|_2^2 \approx f(x_k + s) \quad (4.6)$$

of the smooth term f , to which we add the nonsmooth term h .

The effectiveness of (4.5) comes from the observation that it is possible to determine a closed form solution s_k , or at least a specialized procedure to identify such a global minimizer, for numerous choices of h that are relevant in applications [11], without resort to a general-purpose optimization solver.

Aravkin *et al.* [4] devise a trust-region quasi-Newton method for (6.1) without explicit bound constraints in which subproblems are solved inexactly by a method closely related to (4.5). However, the subproblem has the additional trust-region constraint $s \in \Delta_k \mathbb{B}$:

$$s_k \in \underset{s}{\text{argmin}} \quad \nabla f(x_k)^T s + \frac{1}{2} \nu_k^{-1} \|s\|_2^2 + h(x_k + s) + \chi(s \mid \Delta_k \mathbb{B}), \quad (4.7)$$

where $\Delta_k > 0$, which may be construed as changing $h(x_k + s)$ to $h(x_k + s) + \chi(s \mid \Delta_k \mathbb{B})$. Because both $s \mapsto h(x_k + s)$ and $s \mapsto \chi(s \mid \Delta \mathbb{B})$ are proper and lsc, so is their sum. In addition, because $\Delta_k \mathbb{B}$ is bounded, h no longer needs to be prox-bounded for a solution to exist. Again, a closed form solution may be found for a variety of cases of interest in applications [2, 4].

4.3 The indefinite proximal operator

4.3.1 Definition and Examples

We restrict ourselves to the case where h is separable, i.e., $h(x) = h_1(x_1) + \dots + h_n(x_n)$, in which case the search for an element s of (4.6) decouples componentwise:

$$s_i \in \operatorname{argmin}_{s_i} g_i s_i + \frac{1}{2} \nu^{-1} s_i^2 + h_i(x_i + s_i),$$

where g_i is the i -th component of $\nabla f(x_k)$. If B is defined in the ℓ_∞ -norm, (4.7) also decouples. Without loss of generality, and accounting for the bound constraints of (6.1), we write it as

$$s_i \in \operatorname{argmin}_{s_i} g_i s_i + \frac{1}{2} \nu^{-1} s_i^2 + h_i(x_i + s_i) + \chi(s_i \mid [\tilde{\ell}_i, \tilde{u}_i]), \quad (4.8)$$

where $\tilde{\ell}_i := \max(\ell_i - x_i, -\Delta)$ and $\tilde{u}_i := \min(u_i - x_i, \Delta)$ are the i th components of $\tilde{\ell}$ and $\tilde{u} \in \mathbb{R}^n$, which describe the intersection of the box $[\ell, u]$ with the trust region.

Replacing the quadratic coefficient in (4.8) by a positive scalar $\delta_i > 0$ does not complicate the problem as it simply amounts to changing the value of ν . In the following, we contend that replacing the quadratic coefficient by $\delta_i \leq 0$ continues to allow us to derive a closed form solution in certain cases. Observe that modifying the quadratic term as described amounts to changing (4.6) to

$$f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T D_k s,$$

where $D_k = \operatorname{diag}(d_k)$, and $d_k \in \mathbb{R}^n$ has components $\delta_{k,i}$. Crucially, we wish to allow D_k to be indefinite to capture any nonconvexity of f about x_k to a certain extent. While such change may make the quadratic model nonconvex, the indicator of $[\tilde{\ell}_i, \tilde{u}_i]$ ensures that a finite solution exists.

Definition 1. Let $g \in \mathbb{R}^n$, $D \in \mathbb{R}^{n \times n}$ be diagonal and $C \subset \mathbb{R}^n$ be nonempty. The indefinite proximal operator of h with respect to g , D and C is

$$\operatorname{iprox}(q) := \operatorname{argmin}_{g, D, h, C} g^T x + \frac{1}{2} x^T D x + h(x) + \chi(x \mid C). \quad (4.9)$$

The usual proximal operator of νh at $q \in \mathbb{R}^n$ is a special case of Definition 1 in which $C = \mathbb{R}^n$, $D = \nu^{-1} I$, and $g = -\nu^{-1} q$.

Example 1. Consider $h(x) = \lambda \|x\|_0$, where $\lambda > 0$, and $C := [\ell, u]$, where $\ell, u \in \mathbb{R}^n$ with

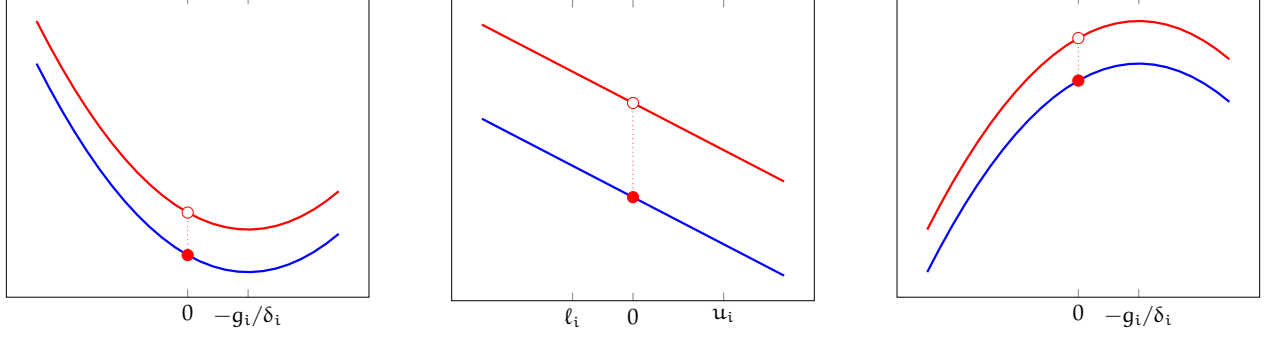


Figure 4.1 Illustration of Example 1 with $\delta_i > 0$ (left), $\delta_i = 0$ (center), and $\delta_i < 0$ (right). The blue curve is $g_i x_i + \frac{1}{2} \delta_i x_i^2$ while the red curve is $g_i x_i + \frac{1}{2} \delta_i x_i^2 + \lambda |x_i|_0$.

$\ell < u$ componentwise. The computation of x in (4.9) decouples componentwise as

$$x_i \in \underset{x_i}{\operatorname{argmin}} g_i x_i + \frac{1}{2} \delta_i x_i^2 + \lambda |x_i|_0 + \chi(x_i \mid [\ell_i, u_i]).$$

Note that

$$g_i x_i + \frac{1}{2} \delta_i x_i^2 + \lambda |x_i|_0 = \begin{cases} 0 & \text{if } x_i = 0 \\ \lambda + g_i x_i + \frac{1}{2} \delta_i x_i^2 & \text{if } x_i \neq 0. \end{cases}$$

If $\delta_i = 0$, the solution set of the above problem is

- $[\ell_i, u_i]$ if $g_i = 0$ and either $\ell_i > 0$ or $u_i < 0$;
- $\operatorname{argmin}\{\lambda + g_i x_i \mid x_i \in M_i\}$ where M_i is the set $\{\ell_i, u_i\}$ together with 0 if $0 \in (\ell_i, u_i)$ in all other cases.

If $\delta_i \neq 0$, the solution set is $\operatorname{argmin}\{\lambda + g_i x_i + \frac{1}{2} \delta_i x_i^2 \mid x_i \in M_i\}$, where M_i is the finite set comprising ℓ_i, u_i together with

- $-g_i/\delta_i$ if $\delta_i > 0$;
- 0 if $0 \in (\ell_i, u_i)$.

See Figure 4.1 for an illustration of a few cases.

Example 2. Consider $h(x) = \lambda \|x\|_1$ where $\lambda > 0$, and C as in Example 1. The computation of x in (4.9) decouples componentwise as

$$x_i \in \underset{x_i}{\operatorname{argmin}} g_i x_i + \frac{1}{2} \delta_i x_i^2 + \lambda |x_i| + \chi(x_i \mid [\ell_i, u_i]).$$

Note that

$$g_i x_i + \frac{1}{2} \delta_i x_i^2 + \lambda |x_i| = \begin{cases} (g_i + \lambda)x_i + \frac{1}{2} \delta_i x_i^2 & \text{if } x_i \geq 0 \\ (g_i - \lambda)x_i + \frac{1}{2} \delta_i x_i^2 & \text{if } x_i < 0. \end{cases}$$

If $\delta_i = 0$, the solution of the above problem is

- $[\ell_i, \max(0, \ell_i)]$ if $g_i = \lambda$;
- $[\min(u_i, 0), u_i]$ if $g_i = -\lambda$;
- $\text{proj}(0 \mid [\ell_i, u_i])$ if $0 \leq |g_i| < \lambda$;
- $\{\ell_i\}$ if $g_i > \lambda$;
- $\{u_i\}$ if $g_i < -\lambda$.

Consider now the case where $\delta_i > 0$. The branch of quadratic for $x_i \geq 0$ lies above the horizontal axis if $g_i \geq -\lambda$, and that for $x_i < 0$ does the same if $g_i \leq \lambda$. Thus, if $|g_i| \leq \lambda$, the unique unconstrained minimizer is 0 and, by convexity, the solution of the constrained problem is $\text{proj}(0 \mid [\ell_i, u_i])$.

If $g_i > \lambda$, for the unconstrained minimizer is $\bar{x}_i := -(g_i - \lambda)/\delta_i < 0$. By convexity, the solution of the constrained problem is $\text{proj}(\bar{x}_i \mid [\ell_i, u_i])$. The situation is similar when $g_i < -\lambda$.

Consider finally the case where $\delta_i < 0$. In this case, the unique constrained minimizer is $\text{argmin}\{g_i x_i + \frac{1}{2}\delta_i x_i^2 + \lambda|x_i| \mid x_i \in M_i\}$ where M_i is the set containing ℓ_i and u_i together with 0 if the latter lies inside $[\ell_i, u_i]$.

Figure 4.2 illustrates a few representative cases.

4.3.2 Properties

At $x \in \mathbb{R}^n$ where h is finite, consider models

$$\varphi(s; x) \approx f(x + s) \tag{4.10a}$$

$$\psi(s; x) \approx h(x + s) \tag{4.10b}$$

$$m(s; x) := \varphi(s; x) + \psi(s; x). \tag{4.10c}$$

We make the following assumption on the models (5.10).

Model Assumption 4.3.1. For any $x \in \mathbb{R}^n$, $\varphi(\cdot; x) \in \mathcal{C}^1$, and satisfies $\varphi(0; x) = f(x)$ and $\nabla\varphi(0; x) = \nabla f(x)$. For any $x \in \mathbb{R}^n$ where h is finite, $\psi(\cdot; x)$ is proper lsc, and satisfies $\psi(0; x) = h(x)$ and $\partial\psi(0; x) = \partial h(x)$.

Proposition 5. Let $C \subset \mathbb{R}^n$ be nonempty and compact, and let Model Assumption 5.2.1 be satisfied. Let (6.1) satisfy the constraint qualification (5.7) at $x \in C$. Assume $0 \in \text{argmin}_s m(s; x) + \chi(x + s \mid C)$, and let the latter subproblem satisfy the constraint qualification (5.7) at $s = 0$. Then x is first-order stationary for (6.1).

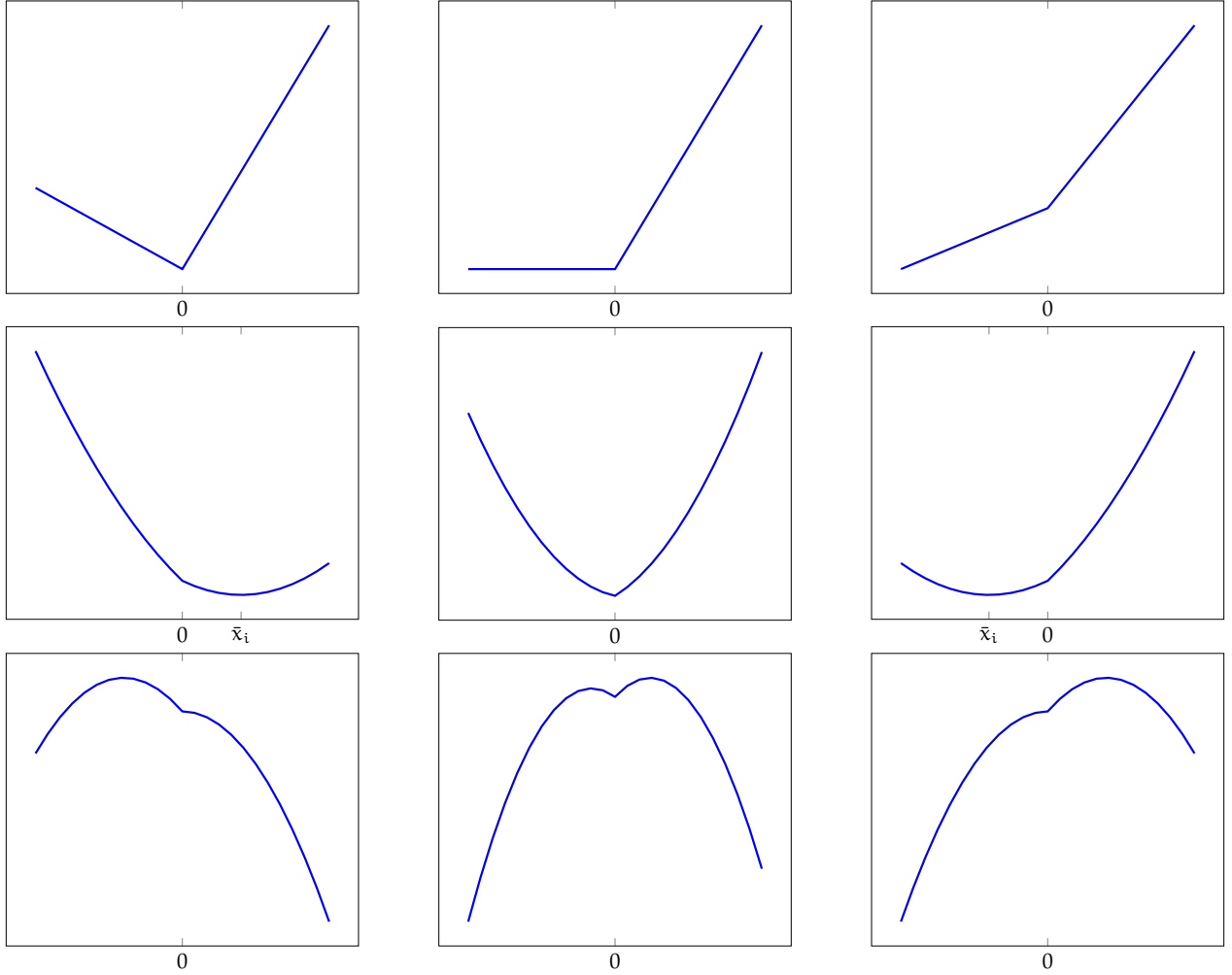


Figure 4.2 Illustration of Example 2. Top row: $\delta_i = 0$. From left to right: $0 \leq g_i < \lambda$, $g_i = \lambda$, and $g_i > \lambda$. Middle row: $\delta_i > 0$. From left to right: $g_i < -\lambda$, $|g_i| \leq \lambda$, and $g_i > \lambda$. Bottom row: $\delta_i < 0$. From left to right: $g_i \leq -\lambda$, $|g_i| < \lambda$, and $g_i > \lambda$.

Proof. Let $\chi_{x,C}(s) := \chi(x + s \mid C)$. Because C is compact, Model Assumption 5.2.1 ensures that $s \mapsto m(s; x) + \chi(x + s \mid C)$ is lsc, bounded below and that a minimizer exists. By assumption, $0 \in \nabla f(x) + \partial(\psi + \chi_{x,C})(0)$. By the constraint qualification for the subproblem and Model Assumption 5.2.1, $0 \in \nabla f(x) + \partial h(x) + N_C(x)$, which are the first-order optimality conditions for (6.1). $\square$

In the assumptions of Proposition 13, satisfaction of the constraint qualification in the subproblem is ensured if $\partial^\infty \psi(0; x) = \partial^\infty h(x)$ provided (5.7) is satisfied at x for (6.1).

4.4 Diagonal quasi-Newton methods

The simplest possible diagonal approximation to $\nabla^2 f(x_k)$ is $B_k := \sigma_k I$ for some scalar σ_k , where I denotes the identity. In the R2 algorithm [4, Algorithm 6.1], σ_k is chosen adaptively at each iteration based on progress. A more sophisticated strategy consists in choosing σ_k so as to best approximate $\nabla^2 f(x_k)$, in a sense to be defined, based on recently observed local information. That is the logic behind the spectral projected gradient method [16], which is easily generalized as a spectral *proximal* gradient method, and where

$$\sigma_k := s_{k-1}^T y_{k-1} / s_{k-1}^T s_{k-1}, \quad s_{k-1} := x_k - x_{k-1}, \quad y_{k-1} := \nabla f(x_k) - \nabla f(x_{k-1}).$$

This choice of σ_k arises from the secant equation $\sigma_k s_{k-1} = y_{k-1}$, which, in general, has no solution, but can be solved in the least-squares sense by minimizing $\|\sigma s_{k-1} - y_{k-1}\|_2$ in terms of σ .

Gilbert and Lemaréchal [46] experiment, among others, with diagonal updates of quasi-Newton approximations. However, there may be no diagonal solution B_k to the secant equation $B_k s_{k-1} = y_{k-1}$, and one must resort to a different approach to update a diagonal approximation. Dennis and Wolkowicz [38] introduce the *weak secant* equation $s_{k-1}^T B_k s_{k-1} = s_{k-1}^T y_{k-1}$ and accompanying update formulae. Nazareth [65] observes that some of those proposed updates can be constrained to update only the diagonal of B_k but that the resulting B_k no longer necessarily satisfies the weak secant equation.

Zhu *et al.* [86] supplement the weak secant equation by additionally requiring that B_k be diagonal, and call the resulting set of conditions the *quasi-Cauchy* conditions. In particular, the spectral gradient approximation is the only solution of the quasi-Cauchy conditions that is a multiple of I . They derive updates from variational principles resembling those used in classic quasi-Newton updates. The first update sets B_k to the unique solution of

$$\underset{B}{\text{minimize}} \quad \|B - B_{k-1}\|_F \quad \text{subject to} \quad s_{k-1}^T B s_{k-1} = s_{k-1}^T y_{k-1}, \quad (4.11)$$

which is the counterpart of the Powell-symmetric-Broyden (PSB) variational problem. The solution of (4.11) is shown to be

$$B_k = B_{k-1} + \frac{s_{k-1}^T (y_{k-1} - B_{k-1} s_{k-1})}{\text{trace}(S_{k-1}^4)} S_{k-1}^2, \quad S_{k-1} := \text{diag}(s_{k-1}). \quad (4.12)$$

Like the PSB update, (4.12) does not possess the hereditary positive-definiteness property. However, capturing negative curvature is a desirable feature in trust-region methods.

Motivated by linesearch methods, Zhu *et al.* [86] devise a second strategy in which they update $B_k^{1/2}$, and that results in hereditary positive definiteness. The main idea is analogous to (4.11) with the objective replaced with $\|B^{1/2} - B_{k-1}^{1/2}\|_F$. The update depends on the root of a scalar nonlinear equation, and is therefore more costly to perform than (4.12).

Andrei [1] suggests an alternative variational problem in which B_k is determined as the solution of

$$\underset{B}{\text{minimize}} \quad \frac{1}{2}\|B - B_{k-1}\|_F^2 + \text{trace}(B) \quad \text{subject to} \quad s_{k-1}^T B s_{k-1} = s_{k-1}^T y_{k-1}, \quad (4.13)$$

where minimizing the trace of B tends to cluster its eigenvalues. The solution of (4.13) is shown to be

$$B_k = B_{k-1} + \frac{s_{k-1}^T (y_{k-1} + s_{k-1} - B_{k-1} s_{k-1})}{\text{trace}(S_{k-1}^4)} S_{k-1}^2 - I. \quad (4.14)$$

Again, (4.14) does not preserve positive definiteness. Also motivated by linesearch methods, Andrei [1] employs a thresholding strategy that ensures computation of a descent direction.

Though Nazareth [65] appears to have been motivated by derivative-free methods,¹ diagonal quasi-Newton updates are good candidate approximations for use in first-order methods for regularized optimization, and we are not aware of any prior work using them.

4.5 The indefinite proximal gradient method

4.5.1 Description of the algorithm and convergence properties

For $\nu > 0$, define

$$\varphi_{\text{cp}}(s; x) := f(x) + \nabla f(x)^T s, \quad (4.15a)$$

$$m(s; x, \nu) := \varphi_{\text{cp}}(s; x) + \frac{1}{2}\nu^{-1}\|s\|_2^2 + \psi(s; x). \quad (4.15b)$$

Guided by Proposition 13, we compute a first step denoted $s_{k,1}$ such that

$$s_{k,1} \in \underset{s}{\text{argmin}} \quad m(s; x_k, \nu_k) + \chi(x_k + s \mid [\ell, u] \cap (x_k + \Delta_k \mathbb{B})), \quad (4.16)$$

for an appropriate value of $\nu_k > 0$.

Let

$$m(s; x, D) := \varphi_{\text{cp}}(s; x) + \frac{1}{2}s^T D s + \psi(s; x), \quad (4.17)$$

1. In the sense that the quasi-Cauchy condition only requires the gradient via $s^T y$.

where D is a diagonal matrix. The indefinite proximal gradient iteration with trust region for (6.1) is defined by changing (5.13) to

$$s_k \in \operatorname{argmin}_s m(s; x_k, D_k) + \chi(x_k + s \mid [\ell, u] \cap (x_k + \Delta_k \mathbb{B})), \quad (4.18)$$

where D_k is diagonal, and by updating x_k and Δ_k as is customary in trust-region methods [34]. Among other possible choices, we focus on the case where D_k results from a diagonal quasi-Newton update.

We summarize the entire procedure as Algorithm 4.5.1, which is a special case of [4, Algorithm 3.1].

Algorithm 4.5.1 Nonsmooth trust-region algorithm with diagonal Hessian.

1: Choose constants

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3 \leq \gamma_4, \quad \text{and} \quad \alpha > 0, \beta \geq 1.$$

2: Choose $x_0 \in \mathbb{R}^n$ where h is finite, $\Delta_0 > 0$, compute $f(x_0) + h(x_0)$.

3: Choose $d_0 \in \mathbb{R}^n$ and set $D_0 = \operatorname{diag}(d_0)$.

4: **for** $k = 0, 1, \dots$ **do**

5: Set $\nu_k := 1/(\|d_k\|_\infty + \alpha^{-1}\Delta_k^{-1})$.

6: Define $\varphi(s; x_k)$ and $\psi(s; x_k)$ according to Model Assumption 5.2.1.

7: Define $m(s; x_k, \nu_k)$ as in (5.12) and compute $s_{k,1}$ as in (5.13).

8: Define $m(s; x_k, D_k)$ as in (5.11) and compute a solution s_k of (4.18) with Δ_k replaced by $\min(\Delta_k, \beta\|s_{k,1}\|)$.

9: Compute the ratio

$$\rho_k := \frac{f(x_k) + h(x_k) - (f(x_k + s_k) + h(x_k + s_k))}{m(0; x_k, D_k) - m(s_k; x_k, D_k)}.$$

10: If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$. Otherwise, set $x_{k+1} = x_k$.

11: Choose $d_{k+1} \in \mathbb{R}^n$ and set $D_{k+1} = \operatorname{diag}(d_{k+1})$.

12: Update the trust-region radius according to

$$\Delta_{k+1} \in \begin{cases} [\gamma_3\Delta_k, \gamma_4\Delta_k] & \text{if } \rho_k \geq \eta_2, & \text{(very successful iteration)} \\ [\gamma_2\Delta_k, \Delta_k] & \text{if } \eta_1 \leq \rho_k < \eta_2, & \text{(successful iteration)} \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k < \eta_1 & \text{(unsuccessful iteration).} \end{cases}$$

In [4, Algorithm 3.1], convergence hinges crucially on the fact that the decrease in $m(\cdot; x_k, D_k)$ achieved by s_k is at least a fraction of the decrease in $m(\cdot; x_k, \nu_k)$ achieved by $s_{k,1}$, the first proximal gradient step with a well-chosen step size $\nu_k > 0$. At iteration k of Algorithm 4.5.1, $\nu_k < 1/\|d_k\|_\infty$. Note that the choice $d_k = 0$ is allowed.

In Algorithm 4.5.1, the computation of $s_{k,1}$ serves two purposes. The first is as stopping condition by Proposition 13, and the second is to set the trust-region radius in the subproblem for s_k . This is at variance with [4, Algorithm 3.1], where, in addition, s_k is computed by continuing the proximal gradient iterations from $s_{k,1}$.

In a similar notation to that of [4], let

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) := f(x_k) + h(x_k) - (\varphi_{\text{cp}}(s_{k,1}; x_k) + \psi(s_{k,1}; x_k)), \quad (4.19a)$$

$$\xi(\Delta_k; x_k, D_k) := f(x_k) + h(x_k) - m(s_k; x_k, D_k) \quad (4.19b)$$

denote the optimal model decrease for (5.12) and (5.11). The definition of (5.13) guarantees that

$$f(x_k) + h(x_k) = m(0; x_k, \nu_k) \geq m(s_{k,1}; x_k, \nu_k) = \varphi_{\text{cp}}(s_{k,1}; x_k) + \psi(s_{k,1}; x_k) + \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2, \quad (4.20)$$

which implies

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2 = \frac{1}{2}(\|d_k\|_\infty + \alpha^{-1}\Delta_k^{-1})\|s_{k,1}\|^2 \geq \frac{1}{2}\alpha^{-1}\Delta_k^{-1}\|s_{k,1}\|^2. \quad (4.21)$$

If $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = 0$, we obtain $s_{k,1} = 0$. Proposition 13 then implies that x_k is first-order stationary provided (6.1) satisfies the constraint qualification at x_k and (5.12) satisfies it at $s = 0$.

To establish convergence properties of Algorithm 4.5.1, we require Aravkin *et al.*'s Step Assumption 3.1, recalled in the following assumption for convenience.

Step Assumption 4.5.1. *There exists $\kappa_m > 0$ and $\kappa_{\text{mdc}} \in (0, 1)$ such that for all k ,*

$$|f(x_k + s_k) + h(x_k + s_k) - (\varphi(s_k; x_k, D_k) + \psi(s_k; x_k))| \leq \kappa_m \|s_k\|_2^2, \quad (4.22a)$$

$$\xi(\Delta_k; x_k, D_k) \geq \kappa_{\text{mdc}} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k). \quad (4.22b)$$

Among other situations, (4.22a) is satisfied if $\{D_k\}$ is bounded, ∇f is Lipschitz continuous, and we select $\psi(s; x_k) = h(x_k + s)$. The following proposition gives a sufficient condition for which (4.22b) is satisfied.

Proposition 6. *If*

$$\|d_k\|_\infty \leq (\kappa_{\text{mdc}}^{-1} - 1)\alpha^{-1}\Delta_k^{-1}, \quad (4.23)$$

then (4.22b) is satisfied.

Proof. By definition of s_k ,

$$\begin{aligned}
m(s_k; x_k, D_k) &\leq m(s_{k,1}; x_k, D_k) \\
&= \varphi(s_{k,1}; x_k) + \frac{1}{2} s_{k,1}^T D_k s_{k,1} + \psi(s_{k,1}; x_k) \\
&\leq \varphi(s_{k,1}; x_k) + \frac{1}{2} \|d_k\|_\infty \|s_{k,1}\|^2 + \psi(s_{k,1}; x_k) \\
&= \varphi(s_{k,1}; x_k) + \frac{1}{2} (\nu_k^{-1} - \alpha^{-1} \Delta_k^{-1}) \|s_{k,1}\|^2 + \psi(s_{k,1}; x_k),
\end{aligned}$$

which leads to

$$\xi(\Delta_k; x_k, D_k) \geq \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) - \frac{1}{2} (\nu_k^{-1} - \alpha^{-1} \Delta_k^{-1}) \|s_{k,1}\|^2.$$

For (4.22b) to be satisfied, it is sufficient to show that

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) - \frac{1}{2} (\nu_k^{-1} - \alpha^{-1} \Delta_k^{-1}) \|s_{k,1}\|^2 \geq \kappa_{\text{mdc}} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k),$$

that we rewrite as

$$(1 - \kappa_{\text{mdc}}) \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2} (\nu_k^{-1} - \alpha^{-1} \Delta_k^{-1}) \|s_{k,1}\|^2. \quad (4.24)$$

Let $\|d_k\|_\infty \leq (\kappa_{\text{mdc}}^{-1} - 1) \alpha^{-1} \Delta_k^{-1}$, which is equivalent to $\nu_k = \frac{1}{\|d_k\|_\infty + \alpha^{-1} \Delta_k^{-1}} \geq \kappa_{\text{mdc}} \alpha \Delta_k$.

Since $\kappa_{\text{mdc}} \in (0, 1)$, we have

$$(1 - \kappa_{\text{mdc}}) \frac{1}{2} \nu_k^{-1} \geq \frac{1}{2} (\nu_k^{-1} - \alpha^{-1} \Delta_k^{-1}).$$

By multiplying by $\|s_{k,1}\|^2$ and recalling that $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2} \nu_k^{-1} \|s_{k,1}\|^2$, we get

$$(1 - \kappa_{\text{mdc}}) \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq (1 - \kappa_{\text{mdc}}) \frac{1}{2} \nu_k^{-1} \|s_{k,1}\|^2 \geq \frac{1}{2} (\nu_k^{-1} - \alpha^{-1} \Delta_k^{-1}) \|s_{k,1}\|^2,$$

so that (4.24) is satisfied. $\square$

We can always choose κ_{mdc} small enough to ensure that $\|d_k\|_\infty$ does not have to be too close to zero if Δ_k does not grow unbounded. Whenever Algorithm 4.5.1 generates infinitely many very successful iterations, we could define a Δ_{max} from which we stop increasing Δ_k . We point out that Proposition 6 only gives a sufficient condition for (4.22b) to be satisfied, but it is not necessary. Therefore, for the rest of this paper, we base our analysis on (4.22b) rather than on (4.23).

Under Step Assumption 4.5.1, we can apply directly Aravkin *et al.*'s convergence properties.

Proposition 7 (4, Theorem 3.4). *Let Step Assumption 4.5.1 be satisfied and*

$$\Delta_{\text{succ}} := \frac{\kappa_{\text{mdc}}(1 - \eta_2)}{2\kappa_{\text{m}}\alpha\beta^2} > 0.$$

If x_k is not first-order stationary and $\Delta_k \leq \Delta_{\text{succ}}$, iteration k is very successful and $\Delta_{k+1} > \Delta_k$.

Without further assumptions, the following result also holds.

Proposition 8 (4, Theorem 3.5). *Let Step Assumption 4.5.1 be satisfied and assume Algorithm 4.5.1 generates a finite number of successful iterations. Then $x_k = x_*$ for all sufficiently large k . If (6.1) satisfies the constraint qualification at x_k and (5.12) satisfies it at $s = 0$, x_* is first-order critical.*

We continue to follow the analysis of [4] and note that $\Delta_k \geq \Delta_{\min}$ for all $k \in \mathbb{N}$, where $\Delta_{\min} := \min(\Delta_0, \gamma_1 \Delta_{\text{succ}})$. Aravkin *et al.* [4] establish that $\xi_{\text{cp}}(\cdot; x_k, \nu_k)$ is increasing, and therefore, $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \xi_{\text{cp}}(\Delta_{\min}; x_k, \nu_k)$. In addition, they show that $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ is an appropriate criticality measure. Let $0 < \epsilon < 1$, and

$$\begin{aligned} I(\epsilon) &:= \{k \in \mathbb{N} \mid \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} > \epsilon\}, \\ S(\epsilon) &:= \{k \in I(\epsilon) \mid \rho_k \geq \eta_1\}, \\ U(\epsilon) &:= \{k \in I(\epsilon) \mid \rho_k < \eta_1\}, \end{aligned}$$

be the set of iterations, successful iterations, and unsuccessful iterations until the criticality measure drops below ϵ , respectively.

We now derive bounds on $|I(\epsilon)|$ using the analysis of [4] specialized to Algorithm 4.5.1 under the assumption that $\{D_k\}$ is bounded. We make the following assumption.

Assumption 1. *There exists $d_{\max} > 0$ such that $\|d_k\|_{\infty} \leq d_{\max}$ for all $k \in \mathbb{N}$.*

In Algorithm 4.5.1, it is not difficult to ensure satisfaction of Assumption 1. For instance, one may prescribe a value $d_{\max} > 0$ and reset each d_k componentwise to $\min(\max(d_k, -d_{\max}), d_{\max})$. We point out that Assumption 1 holds if (4.23) in Proposition 6 is satisfied for all k , because, in this case, $\|d_k\|_{\infty} \leq d_{\max} := (\kappa_{\text{mdc}}^{-1} - 1)\alpha^{-1}\Delta_{\min}^{-1}$.

Under Assumption 1, the choice of ν_k in Algorithm 4.5.1 guarantees that

$$\nu_k \geq \nu_{\min} := 1/(d_{\max} + \alpha^{-1}\Delta_{\min}^{-1}) > 0.$$

The following result establishes that $|S(\epsilon)|$ is $O(\epsilon^{-2})$.

Proposition 9 (4, Lemma 3.6). *Let Step Assumption 4.5.1 and Assumption 1 be satisfied. Assume that Algorithm 4.5.1 generates infinitely many successful iterations and that there exists $(f + h)_{\text{low}} \in \mathbb{R}$ such that $(f + h)(x_k) \geq (f + h)_{\text{low}}$ for all $k \in \mathbb{N}$. Let $\epsilon \in (0, 1)$. Then,*

$$|S(\epsilon)| \leq \frac{(f + h)(x_0) - (f + h)_{\text{low}}}{\eta_1 \kappa_{\text{mdc}} \nu_{\text{min}} \epsilon^2}.$$

The next result establishes that $|U(\epsilon)|$ is also $O(\epsilon^{-2})$.

Proposition 10 (4, Lemma 3.7). *Under the assumptions of Proposition 9,*

$$|U(\epsilon)| \leq \log_{\gamma_2}(\Delta_{\text{min}}/\Delta_0) + |S(\epsilon)| |\log_{\gamma_2}(\gamma_4)|.$$

The direct consequence of Propositions 9 and 10 is that $|I(\epsilon)|$ is also $O(\epsilon^{-2})$. Because $\epsilon \in (0, 1)$ is arbitrary, we conclude that $\liminf \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = 0$.

4.5.2 A variant saving proximal operator computations

We present a variant of Algorithm 4.5.1 in which we set

$$\nu_k := 1/(\|d_k\|_\infty + \alpha^{-1}) \quad (4.25)$$

in line 5, we do not compute $s_{k,1}$ in line 7, and we leave Δ_k unchanged in line 8, thus saving a proximal operator computation at each iteration. Under Assumption 1, this choice of ν_k leads to a new ν_{min}

$$\nu_k \geq \nu_{\text{min}} := 1/(d_{\text{max}} + \alpha^{-1}) > 0. \quad (4.26)$$

We use the criticality measure

$$\nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2}, \quad (4.27)$$

and modify the definition of $I(\epsilon)$, $S(\epsilon)$ and $U(\epsilon)$ accordingly. To establish the same convergence properties as those of Algorithm 4.5.1, we need Step Assumption 4.5.1 to hold for our algorithmic variant. The following assumption summarizes our requirements.

Step Assumption 4.5.2. *There exists $\kappa_{\text{m}} > 0$ and $\kappa_{\text{mdc}} \in (0, 1)$ such that for all k ,*

$$|f(x_k + s_k) + h(x_k + s_k) - (\varphi(s_k; x_k, D_k) + \psi(s_k; x_k))| \leq \kappa_{\text{m}} \|s_k\|_2^2, \quad (4.28a)$$

$$\xi(\Delta_k; x_k, D_k) \geq \kappa_{\text{mdc}} \xi_{\text{cp}}(\Delta_k; x_k; \nu_k), \quad (4.28b)$$

where $\nu_k = 1/(\|d_k\|_\infty + \alpha^{-1})$ in line 5 and s_k is computed without changing Δ_k in line 8 of

Algorithm 4.5.1.

Even though Step Assumption 4.5.2 involves $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)$, we do not need to compute it in practice. Conveniently, if $\{D_k\}$ is bounded, ∇f is Lipschitz continuous, and we select $\psi(s; x_k) = h(x_k + s)$, (4.28a) is still satisfied. The following proposition, analogous to Proposition 6, shows that (4.29) is a sufficient condition for (4.28b) to hold.

Proposition 11. *If*

$$\|d_k\|_\infty \leq (\kappa_{\text{mdc}}^{-1} - 1)\alpha^{-1}, \quad (4.29)$$

then (4.28b) is satisfied for Algorithm 4.5.1 with $\nu_k = 1/(\|d_k\|_\infty + \alpha^{-1})$ in line 5, $s_{k,1}$ is not computed in line 7, and Δ_k is unchanged in line 8.

Proof. The proof is identical to that of Proposition 6 if we use (4.25). $\square$

Whenever (4.28b) holds, we have

$$\nu_k^{-1/2} \xi(\Delta_k; x_k, \nu_k)^{1/2} \geq \kappa_{\text{mdc}}^{1/2} \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} \geq \kappa_{\text{mdc}}^{1/2} \nu_k^{-1} \|s_{k,1}\|, \quad (4.30)$$

which ensures that $\nu_k^{-1/2} \xi(\Delta_k; x_k, \nu_k)^{1/2} \rightarrow 0$ implies $\|s_{k,1}\| \rightarrow 0$, and that $\nu_k^{-1/2} \xi(\Delta_k; x_k, \nu_k)^{1/2}$ is an appropriate criticality measure. We emphasize that there is no need to compute $s_{k,1}$ and $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)$; (4.30) is only used as a theoretical justification for the choice of our criticality measure.

We now establish convergence properties similar to those of Section 4.5.1, and follow the analysis of Cartis *et al.* [30, Chapter 2.3].

Proposition 12. *Let Assumption 1 and Step Assumption 4.5.2 be satisfied. Let $\nu_{\min}$ be as defined in (4.26). If*

$$\Delta_k \leq \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_{\text{m}}}} \nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2} \quad (4.31)$$

and x_k is not first-order stationary, iteration k is very successful and $\Delta_{k+1} > \Delta_k$.

Proof. Using Step Assumption 4.5.2 and $\|s_k\| \leq \Delta_k$, we have

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k + s_k) + h(x_k + s_k) - m(s_k; x_k, D_k)}{m(0; x_k, D_k) - m(s_k; x_k, D_k)} \right| \\ &\leq \frac{\kappa_{\text{m}} \|s_k\|^2}{\xi(\Delta_k; x_k, D_k)} \\ &\leq \frac{\kappa_{\text{m}} \Delta_k^2}{\xi(\Delta_k; x_k, D_k)}. \end{aligned}$$

Assumption 1 and (4.26) lead to

$$|\rho_k - 1| \leq \frac{\kappa_m \nu_{\min}^{-1} \Delta_k^2}{\nu_k^{-1} \xi(\Delta_k; x_k, D_k)}.$$

Whenever (4.31) holds, we have $\rho_k \geq \eta_2$, implying that iteration k is very successful and $\Delta_{k+1} \geq \Delta_k$. $\square$

The following lemma is inspired by Conn *et al.* [34, Theorem 6.4.3] and Cartis *et al.* [30, Lemma 2.3.4].

Lemma 1. *Let Step Assumption 4.5.2 be satisfied. Then, for all $k \geq 0$,*

$$\Delta_k > \gamma_1 \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_m}} \min \left(1, \Delta_0 \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \min_{i \in [0, k]} \nu_i^{-1/2} \xi(\Delta_i; x_i, D_i)^{1/2}. \quad (4.32)$$

Proof. We proceed as in Cartis *et al.* [30, Lemma 2.3.4]. The bound certainly holds for $k = 0$, because

$$\Delta_0 > \gamma_1 \sqrt{1 - \eta_2} \Delta_0.$$

Now, we proceed by contradiction and assume that $k \geq 1$ is the first iteration such that (4.32) is not verified.

$$\begin{aligned} \Delta_k &\leq \gamma_1 \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_m}} \min \left(1, \Delta_0 \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \min_{i \in [0, k]} \nu_i^{-1/2} \xi(\Delta_i; x_i, D_i)^{1/2} \\ &\leq \gamma_1 \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_m}} \nu_{k-1}^{-1/2} \xi(\Delta_{k-1}; x_{k-1}, D_{k-1})^{1/2}. \end{aligned}$$

We have $\gamma_1 \Delta_{k-1} \leq \Delta_k$ because of the updating rules of Δ_k in Algorithm 4.5.1, which implies that

$$\Delta_{k-1} \leq \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_m}} \nu_{k-1}^{-1/2} \xi(\Delta_{k-1}; x_{k-1}, D_{k-1})^{1/2}.$$

Using Proposition 12, this results in iteration $k - 1$ being very successful and $\Delta_k > \Delta_{k-1}$. Therefore,

$$\begin{aligned} \Delta_{k-1} &< \Delta_k \\ &\leq \gamma_1 \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_m}} \min \left(1, \Delta_0 \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \min_{i \in [0, k]} \nu_i^{-1/2} \xi(\Delta_i; x_i, D_i)^{1/2} \\ &\leq \gamma_1 \sqrt{\frac{\nu_{\min}(1 - \eta_2)}{\kappa_m}} \min \left(1, \Delta_0 \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \min_{i \in [0, k-1]} \nu_i^{-1/2} \xi(\Delta_i; x_i, D_i)^{1/2}, \end{aligned}$$

which is a contradiction with iteration k being the first to violate (4.32). $\square$

Now, we show analogous properties to Proposition 9 and Proposition 10 for our variant.

Lemma 2. *Let Step Assumption 4.5.2 and Assumption 1 be satisfied. Assume that Algorithm 4.5.1 with ν_k as in (4.25) in line 5, without line 7, and with Δ_k unchanged in line 8 generates infinitely many successful iterations and that there exists $(f + h)_{\text{low}} \in \mathbb{R}$ such that $(f + h)(x_k) \geq (f + h)_{\text{low}}$ for all $k \in \mathbb{N}$. Let $\epsilon \in (0, 1)$. Then,*

$$|S(\epsilon)| \leq \frac{(f + h)(x_0) - (f + h)_{\text{low}}}{\eta_1 \nu_{\min} \epsilon^2}.$$

Proof. For $k \in S(\epsilon)$, we have

$$\begin{aligned} f(x_k) + h(x_k) - f(x_k + s_k) - h(x_k + s_k) &\geq \eta_1(m(0; x_k, D_k) - m(s_k; x_k, D_k)) \\ &= \eta_1 \xi(\Delta_k; x_k, D_k) \\ &\geq \eta_1 \nu_k \epsilon^2 \\ &\geq \eta_1 \nu_{\min} \epsilon^2. \end{aligned}$$

As $(f + h)(x_k) \geq (f + h)_{\text{low}}$, we sum the above inequalities for $k \in S(\epsilon)$ and get

$$(f + h)(x_0) - (f + h)_{\text{low}} \geq \sum_{k \in S(\epsilon)} (f + h)(x_k) - (f + h)(x_{k+1}) \geq |S(\epsilon)| \eta_1 \epsilon^2 \nu_{\min}. \quad \square$$

Lemma 3. *Under the assumptions of Lemma 2,*

$$|U(\epsilon)| = O(\epsilon^{-2}).$$

Proof. Let $k < k(\epsilon)$. Because

$$\nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2} > \epsilon,$$

and

$$\nu_{k(\epsilon)}^{-1/2} \xi(\Delta_{k(\epsilon)}; x_{k(\epsilon)}, D_{k(\epsilon)})^{1/2} \leq \epsilon,$$

we have with Lemma 1

$$\begin{aligned}\Delta_{k(\epsilon)-1} &> \gamma_1 \sqrt{\frac{\nu_{\min}(1-\eta_2)}{\kappa_m}} \min \left(1, \Delta_0 \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \min_{i \in [0, k]} \nu_i^{-1} \xi(\Delta_i; x_i, D_i) \\ &> \gamma_1 \sqrt{\frac{\nu_{\min}(1-\eta_2)}{\kappa_m}} \min \left(1, \Delta_0 \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \epsilon.\end{aligned}$$

For each successful iteration, $\Delta_{k+1} \leq \gamma_4 \Delta_k$, and for each unsuccessful iteration, $\Delta_{k+1} \leq \gamma_2 \Delta_k$, which implies

$$\Delta_{k(\epsilon)-1} \leq \Delta_0 \gamma_2^{|U(\epsilon)|} \gamma_4^{|S(\epsilon)|}.$$

By taking the natural logarithm of the above inequality, we have

$$\begin{aligned}|U(\epsilon)| \log(\gamma_2) + |S(\epsilon)| \log(\gamma_4) &\geq \log(\Delta_{k(\epsilon)-1} / \Delta_0) \\ &\geq \log \left(\gamma_1 \sqrt{\frac{\nu_{\min}(1-\eta_2)}{\kappa_m}} \min \left(\Delta_0^{-1}, \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \epsilon \right),\end{aligned}$$

so that, using the previous inequalities and Lemma 2

$$\begin{aligned}|U(\epsilon)| &\leq \log_{\gamma_2} \left(\gamma_1 \sqrt{\frac{\nu_{\min}(1-\eta_2)}{\kappa_m}} \min \left(\Delta_0^{-1}, \frac{\sqrt{\kappa_m \nu_{\min}^{-1}}}{\nu_0^{-1/2} \xi(\Delta_0; x_0, D_0)^{1/2}} \right) \epsilon \right) + |S(\epsilon)| |\log_{\gamma_2}(\gamma_4)| \\ &\leq O(|\log_{\gamma_2}(\epsilon)|) + O(\epsilon^{-2}) \\ &= O(\epsilon^{-2}).\end{aligned}$$

□

Lemma 2 and Lemma 3 also indicate that $I(\epsilon)$ is $O(\epsilon^{-2})$, and $\liminf \nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2} = 0$. With (4.28b) in Step Assumption 4.5.2, we also have $\liminf \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = 0$.

Finally, we emphasize the changes made in this section to Algorithm 4.5.1:

- in line 5, $\nu_k = 1/(\|d_k\|_\infty + \alpha^{-1} \Delta_k^{-1})$ became $\nu_k = 1/(\|d_k\|_\infty + \alpha^{-1})$,
- line 7 of Algorithm 4.5.1 was removed,
- Δ_k was left unchanged in line 8 of Algorithm 4.5.1,
- we used $\nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2}$ as a criticality measure, instead of $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)$, however, when (4.28a) holds, (4.30) indicates that if the new criticality measure is small, then the criticality measure of Section 4.5.1 is also small,
- in Proposition 12, there is no $\Delta_{\min} > 0$ independent of k as in Proposition 24,
- under the assumptions of Lemma 2, which are similar to those of Proposition 9, the

complexity bound is still in $O(\epsilon^{-2})$, and

$$\liminf \nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2} = \liminf \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = 0.$$

4.6 Implementation and numerical experiments

Our Julia implementation of Algorithm 4.5.1 is available from our RegularizedOptimization package [10] under the name TRDH. The latter can be used to solve (6.1) directly, or as subproblem solver in TR [4, Algorithm 3.1], instead of R2 [4, Algorithm 6.1]. Below, we use the notation TR-R2 and TR-TRDH to denote the application of TR to solve (6.1) with R2 or TRDH as subproblem solver, respectively.

In our experiments, TR uses either an LSR1 or an LBFGS quasi-Newton Hessian approximation with memory 5, as implemented in the LinearOperators package [70]. The same package implements diagonal quasi-Newton operators in TRDH using the spectral update, the PSB update (4.12), and the update of Andrei [1] (4.14) with the modification that we scaled the weak secant equations as

$$\tilde{s}_{k-1}^T B \tilde{s}_{k-1} = \tilde{s}_{k-1}^T \tilde{y}_{k-1} \quad (4.34)$$

in (4.11) and (4.13), where $\tilde{s}_{k-1} := s_{k-1} / \|s_{k-1}\|_2$, and $\tilde{y}_{k-1} := y_{k-1} / \|s_{k-1}\|_2$, in order to alleviate numerical issues as s_{k-1} approaches zero.

The indefinite proximal operators of Example 1 and Example 2 are implemented as part of the ShiftedProximalOperators package [9].

When using TRDH as the main solver, we initialize $D_0 := \nu_0^{-1} I$ for $\nu_0 > 0$ given below. When using TR-TRDH with a spectral diagonal quasi-Newton approximation, denoted TR-TRDH-Spec, we set the initial diagonal Hessian approximation in TRDH at iteration k of TR to $D_{k,0} := \nu_k^{-1} I$ (as we would initialize R2 in TR-R2). When using TR-TRDH with the PSB or the Andrei quasi-Newton approximations, denoted TR-TRDH-PSB and TR-TRDH-Andrei, respectively, we set $D_{k,0} := \text{diag}(B_k)$, where B_k is the quasi-Newton Hessian approximation at iteration k of TR.

We set $\psi(s; x_k) := h(x_k + s)$. We initialize $\nu_0 = 1$ for R2 and TRDH used by themselves. The stopping criteria that we used for TR, TRDH and R2 (as subproblem solvers or main solvers) are based on $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$. We set $\Delta_0 = 1$ for TR and for TRDH used as main solver. For TR-TRDH, at iteration k of TR, the initial value of the TRDH trust-region radius is $\Delta_{k,0} = \min(\Delta_k, \beta \|s_{k,1}\|) / 10$, where Δ_k is the TR trust-region radius at iteration k , and $s_{k,1}$ is the first step of the k -th TR subproblem. In other words, the initial TRDH trust-region radius to solve the k -th TR subproblem is a tenth of the trust-region radius of this k -th

subproblem.

For all solvers, the outer iterations terminate as soon as

$$\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} < \epsilon_a + \epsilon_r \nu_0^{-1/2} \xi_{\text{cp}}(\Delta_0; x_0, \nu_0)^{1/2}, \quad (4.35)$$

where $\epsilon_a > 0$ and $\epsilon_r > 0$ are an absolute and a relative tolerance. A round of inner iterations in TR terminates as soon as the stationarity measure of the inner solver satisfies (4.35), with $\tilde{\epsilon}_a = 10^{-5}$ for the first TR iteration, otherwise

$$\tilde{\epsilon}_a = \max(\epsilon_{a,i}, \min(10^{-2}, \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2})),$$

and $\tilde{\epsilon}_r = \epsilon_{r,i}$, where $\epsilon_{a,i} > 0$ and $\epsilon_{r,i} > 0$ are some absolute and relative inner tolerances, and $s_{k,1}$ is the first iterate of the solution of the trust-region subproblem. In the experiments below, we use $\epsilon_{a,i} = 10^{-3}$ and $\epsilon_{r,i} = 10^{-6}$, except in Section 4.6.1, where we use $\epsilon_{a,i} = 10^{-5}$.

Additionally, we test the variant presented in Section 4.5.2, which is denoted “iTRDH” (*indefinite trust-region with diagonal Hessian approximations*) in our results. When in use, the stopping criterion is based on $\nu_k^{-1/2} \xi(\Delta_k; x_k, D_k)^{1/2}$, instead of $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$.

In our results, we report

- the final $f(x)$;
- the final $h(x)/\lambda$;
- the final stationarity measure $\sqrt{\xi/\nu}$;
- $\|x - x_\star\|_2$, where $x_\star$ is the exact solution, if it is available;
- the number of smooth objective evaluations $\# f$;
- the number of gradient evaluations $\# \nabla f$;
- the number of proximal operator evaluations $\# \text{prox}$;
- the elapsed time t in seconds.

Because our implementations are not yet perfectly optimized in terms of memory allocations, we neglect the elapsed time in our interpretations, and only report it in the tables as an indicator for the reader.

In our test cases, available from the RegularizedProblems package [8], the computational cost of evaluating the gradient is significantly higher than a proximal evaluation or an objective evaluation. In all cases, except in case of failure, all solvers find similar final solutions, and we only show one for illustration.

4.6.1 Basis pursuit denoise (BPDN)

Our first test case is the basis pursuit denoise (BPDN) problem [39, 78]. The stopping tolerances ϵ_a and ϵ_r are set to 10^{-5} . In this subsection only, when using TR, we set $\epsilon_{a,i} = 10^{-5}$ in order to have accurate subproblem solves, which leads to performing fewer gradient evaluations without sacrificing too many proximal evaluations. Let $m = 200$, $n = 512$, $b = Ax_\star + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.01)$, $A \in \mathbb{R}^{m \times n}$ has orthonormal rows, and $x_\star$ is a vector of zeros, except for 10 of its components that are set to ± 1 . We solve

$$\underset{x}{\text{minimize}} \frac{1}{2} \|Ax - b\|_2^2 + h(x), \quad (4.36)$$

where $h(x) = \lambda \|x\|_0$. As in [4], we use $\lambda = 0.1 \|A^T b\|_\infty$.

Table 4.1 BPDN (4.36) statistics with $h = \lambda \|\cdot\|_0$. All variants of TR use an LSR1 Hessian approximation, and are given a maximum of 100 inner iterations. The optimal objective value is $f(x_\star) = 9.90e - 03$.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\ x - x_\star\ _2$	$\#f$	$\#\nabla f$	$\# \text{prox}$	$t \text{ (s)}$
R2	9.44e-03	10	4.4e-03	4.7e-02	30	31	30	8.0e-03
TRDH-Spec	9.44e-03	10	4.0e-03	4.7e-02	9	9	17	1.7e-02
iTRDH-Spec	9.44e-03	10	3.4e-03	4.7e-02	10	9	9	4.0e-03
TRDH-PSB	9.44e-03	10	3.7e-03	4.7e-02	18	15	35	1.1e-02
iTRDH-PSB	9.44e-03	10	2.6e-03	4.7e-02	20	16	19	5.0e-03
TRDH-Andrei	9.44e-03	10	4.5e-03	4.7e-02	48	29	95	1.0e-02
iTRDH-Andrei	9.44e-03	10	3.7e-03	4.7e-02	59	35	58	3.1e-02
TR-R2	9.44e-03	10	1.7e-05	4.7e-02	23	23	40	1.8e-02
TR-TRDH-PSB	9.44e-03	10	1.6e-05	4.7e-02	21	21	58	6.9e-02
TR-iTRDH-PSB	9.44e-03	10	1.6e-05	4.7e-02	21	21	39	2.5e-02
TR-TRDH-Andrei	9.44e-03	10	1.6e-05	4.7e-02	20	20	127	2.5e-02
TR-iTRDH-Andrei	9.44e-03	10	1.4e-05	4.7e-02	20	20	100	3.0e-02
TR-TRDH-Spec	9.44e-03	10	1.9e-05	4.7e-02	21	21	46	1.8e-02
TR-iTRDH-Spec	9.44e-03	10	1.9e-05	4.7e-02	21	21	33	1.8e-02

Figure 4.3 shows the solution of (4.36) with TRDH-Spec. Table 4.1 reports the statistics for the different solvers and shows that TRDH-Spec and TRDH-PSB perform fewer objective and gradient evaluations than R2. TRDH-Andrei performs worse than R2 on this problem. As expected, the “iTRDH” variants require fewer proximal operator evaluations. They result in similar numbers of objective and gradient evaluations in the cases of TRDH-Spec and TRDH-PSB, but require more evaluations in the case of TRDH-Andrei.

All TR-TRDH and TR-iTRDH solvers perform fewer objective and gradient evaluations than TR-R2.

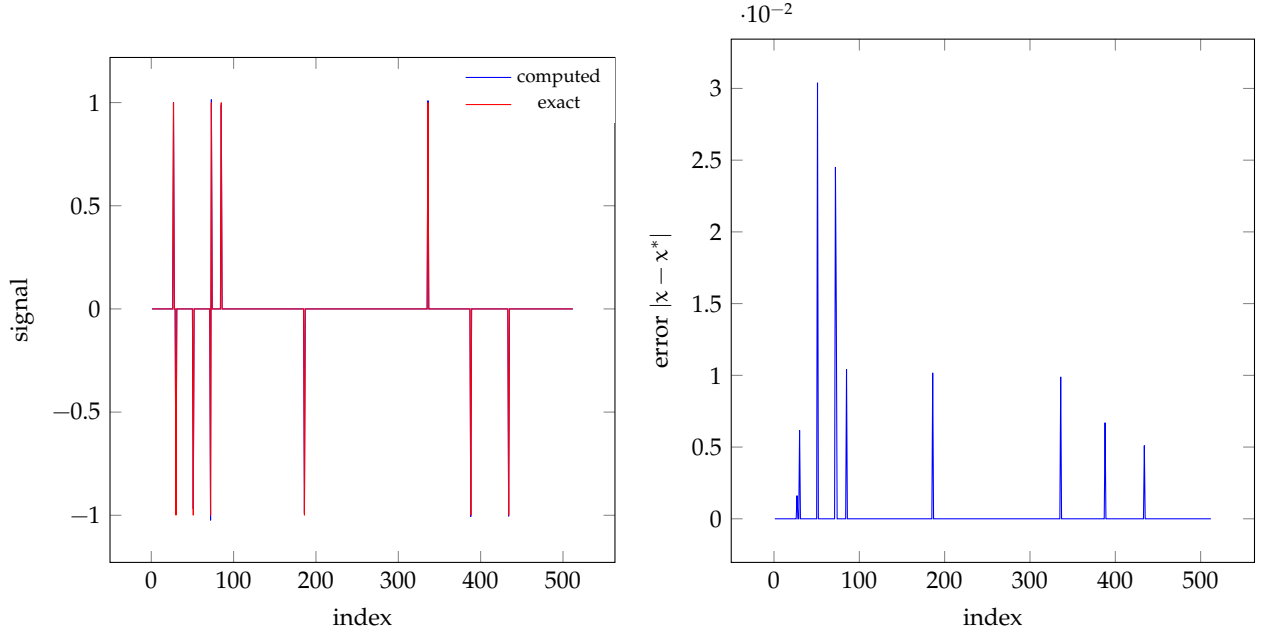


Figure 4.3 Solution of (4.36) (left), and error (right) with TRDH-Spec.

We now solve the constrained variant

$$\underset{x}{\text{minimize}} \frac{1}{2} \|Ax - b\|_2^2 + h(x) \quad \text{subject to } x \geq 0, \quad (4.37)$$

where each element in x_* is either 0 or 1.

Figure 4.4 shows the solution of (6.87) with TRDH-Spec, and Table 6.4 reports the statistics for the different solvers. We observe that TRDH-Spec, iTRDH-Spec, TRDH-PSB and iTRDH-PSB used as main solvers outperform R2 in terms of number of objective and gradient evaluations. TR-TRDH and TR-iTRDH perform similarly to TR-R2.

4.6.2 Sparse nonnegative matrix factorization (NNMF)

Our next test case is a variant of the NNMF problem of Kim and Park [52]. Let $A \in \mathbb{R}^{m \times n}$ have nonnegative entries, where each row represents a feature and each column represents an observation. We wish to factorize $A \approx WH$ by separating A into $k < \min(m, n)$ clusters, where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$ both have nonnegative entries and H is sparse. The problem is stated as

$$\underset{W, H}{\text{minimize}} \frac{1}{2} \|A - WH\|_F^2 + h(H) \quad \text{subject to } W, H \geq 0, \quad (4.38)$$

Table 4.2 Constrained BPDN (6.87) statistics with $h = \lambda \|\cdot\|_0$. All variants of TR use an LSR1 Hessian approximation, and are given a maximum of 100 inner iterations. The optimal objective value is $f(x_*) = 8.98e - 03$.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\ x - x_*\ _2$	$\#f$	$\#\nabla f$	$\# \text{prox}$	$t \text{ (s)}$
R2	8.70e-03	10	3.9e-03	3.6e-02	27	28	27	1.9e-02
TRDH-Spec	8.70e-03	10	3.6e-03	3.6e-02	8	8	15	4.0e-03
iTRDH-Spec	8.70e-03	10	3.0e-03	3.6e-02	9	8	8	2.0e-03
TRDH-PSB	8.70e-03	10	3.6e-03	3.6e-02	11	11	21	3.0e-03
iTRDH-PSB	8.70e-03	10	3.7e-03	3.6e-02	12	11	11	3.0e-03
TRDH-Andrei	8.70e-03	10	4.3e-03	3.6e-02	48	29	95	1.6e-02
iTRDH-Andrei	8.70e-03	10	4.0e-03	3.6e-02	59	35	58	2.0e-02
TR-R2	8.70e-03	10	1.9e-05	3.6e-02	20	20	32	1.2e-02
TR-TRDH-PSB	8.70e-03	10	2.1e-05	3.6e-02	19	19	50	1.8e-02
TR-iTRDH-PSB	8.70e-03	10	2.1e-05	3.6e-02	19	19	34	2.5e-02
TR-TRDH-Andrei	8.70e-03	10	1.9e-05	3.6e-02	20	20	63	1.8e-02
TR-iTRDH-Andrei	8.70e-03	10	1.5e-05	3.6e-02	20	20	51	7.0e-03
TR-TRDH-Spec	8.70e-03	10	2.0e-05	3.6e-02	19	19	28	1.0e-02
TR-iTRDH-Spec	8.70e-03	10	2.0e-05	3.6e-02	19	19	23	1.2e-02

where $h(H) = \lambda \|\text{vec } H\|_0$ and vec stacks the columns of a matrix to form a vector. In our experiments, each observation is generated using a mixture of Gaussians. Negative elements in the matrix A generated are reset to zero.

Table 4.3 NMF (6.84) statistics with $h = \lambda \|\cdot\|_0$. All variants of TR use an LSR1 Hessian approximation, and are given a maximum of 100 inner iterations.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\#f$	$\#\nabla f$	$\# \text{prox}$	$t \text{ (s)}$
R2	2.84e+03	0	2.2e-04	2	2	2	0.0e+00
TRDH-Spec	1.25e+02	50	7.7e-02	47	29	93	5.0e-03
iTRDH-Spec	1.25e+02	50	6.5e-02	48	29	47	4.0e-03
TRDH-PSB	1.70e+02	62	5.7e+00	501	352	1000	5.6e-02
iTRDH-PSB	1.70e+02	62	8.4e+00	501	352	500	5.5e-02
TRDH-Andrei	2.73e+02	54	5.9e+00	501	283	1000	5.9e-02
iTRDH-Andrei	2.73e+02	54	7.0e+00	501	283	500	5.6e-02
TR-R2	1.25e+02	50	5.0e-03	212	115	4983	2.4e-01
TR-TRDH-PSB	1.25e+02	50	7.2e-03	197	104	9826	7.3e-01
TR-iTRDH-PSB	1.25e+02	50	6.4e-03	129	71	3759	4.8e-01
TR-TRDH-Andrei	1.25e+02	50	2.9e-03	170	91	8746	6.1e-01
TR-iTRDH-Andrei	1.25e+02	50	6.8e-03	152	83	4809	6.1e-01
TR-TRDH-Spec	1.25e+02	50	4.8e-03	126	69	4793	3.6e-01
TR-iTRDH-Spec	1.25e+02	50	3.6e-03	162	82	2503	3.5e-01

We set $m = 100$, $n = 50$, $k = 5$, $\lambda = 10^{-1}$. The stopping tolerances ϵ_a and ϵ_r are set to 10^{-5} . Figure 4.5 shows the solution of (6.84) with TR-TRDH-PSB. The statistics are reported in Table 6.2. We observe that R2 is trapped in a spurious stationary point, and that all TRDH and iTRDH solvers except TRDH-Spec and iTRDH-Spec reach their maximum number of

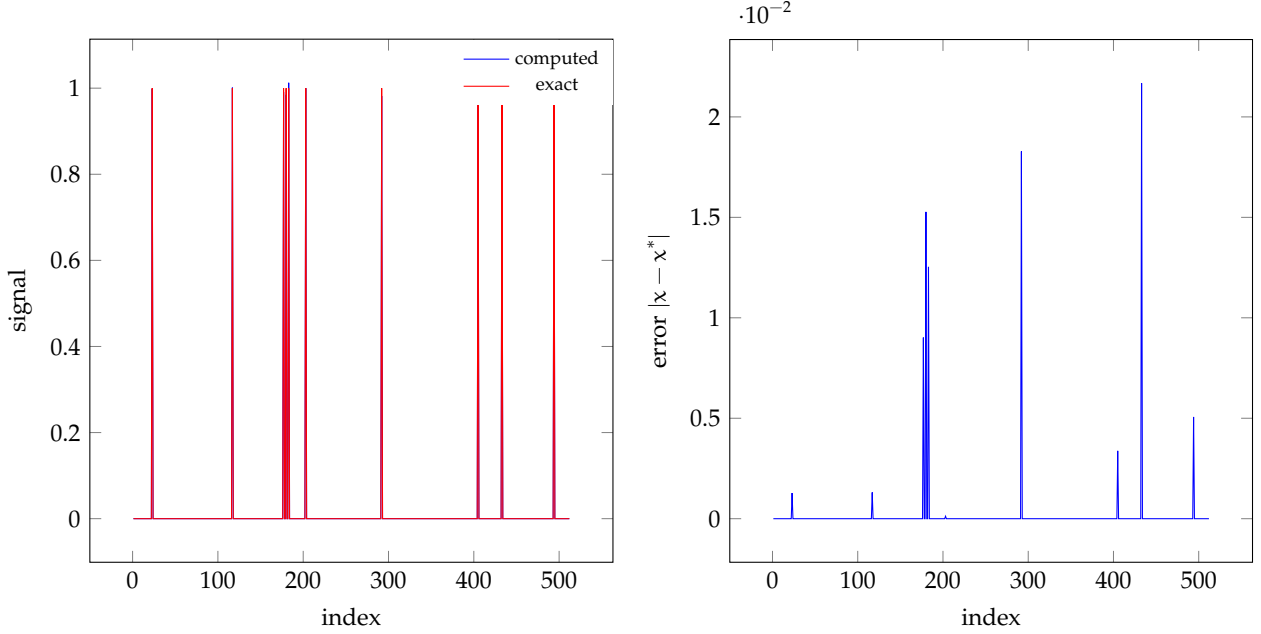


Figure 4.4 Solution of (6.87) (left), and error (right) with TR-TRDH-Spec.

iterations. All solvers using TR perform well with fewer objective and gradient evaluations than TR-R2. The number of proximal operator calls is lower for all TR-iTRDH variants.

4.6.3 Nonlinear support vector machine (SVM)

We now consider the nonlinear SVM described by [2] to classify digits of the MNIST dataset as either 1 or 7—the other digits are removed. Let m be the number of images and n the vectorized image size. The problem reads

$$\underset{x}{\text{minimize}} \frac{1}{2} \|\mathbf{1} - \tanh(b \odot (A^T x))\|^2 + h(x), \quad (4.39)$$

where $A \in \mathbb{R}^{m \times n}$ is the data matrix, b is the vector of labels with values ± 1 for all its components, $\odot$ denotes the elementwise product between two vectors, and $h(x) = \lambda \|x\|_1$. We use $n = 784$, $m = 13007$ in the training set, $m = 2163$ in the testing set, $\lambda = 0.1$ and we initialize the problem at $x = e$, the vector of ones. We set the absolute and relative stopping tolerances ϵ_a and ϵ_r to 10^{-4} .

The solution map of (4.39) representing the most important pixels to classify the images between 1 and 7 is shown in Figure 4.6 for the solver TR-TRDH-Spec, along with a few sample digits. We observe that the mid-height pixels are selected as the most important ones,

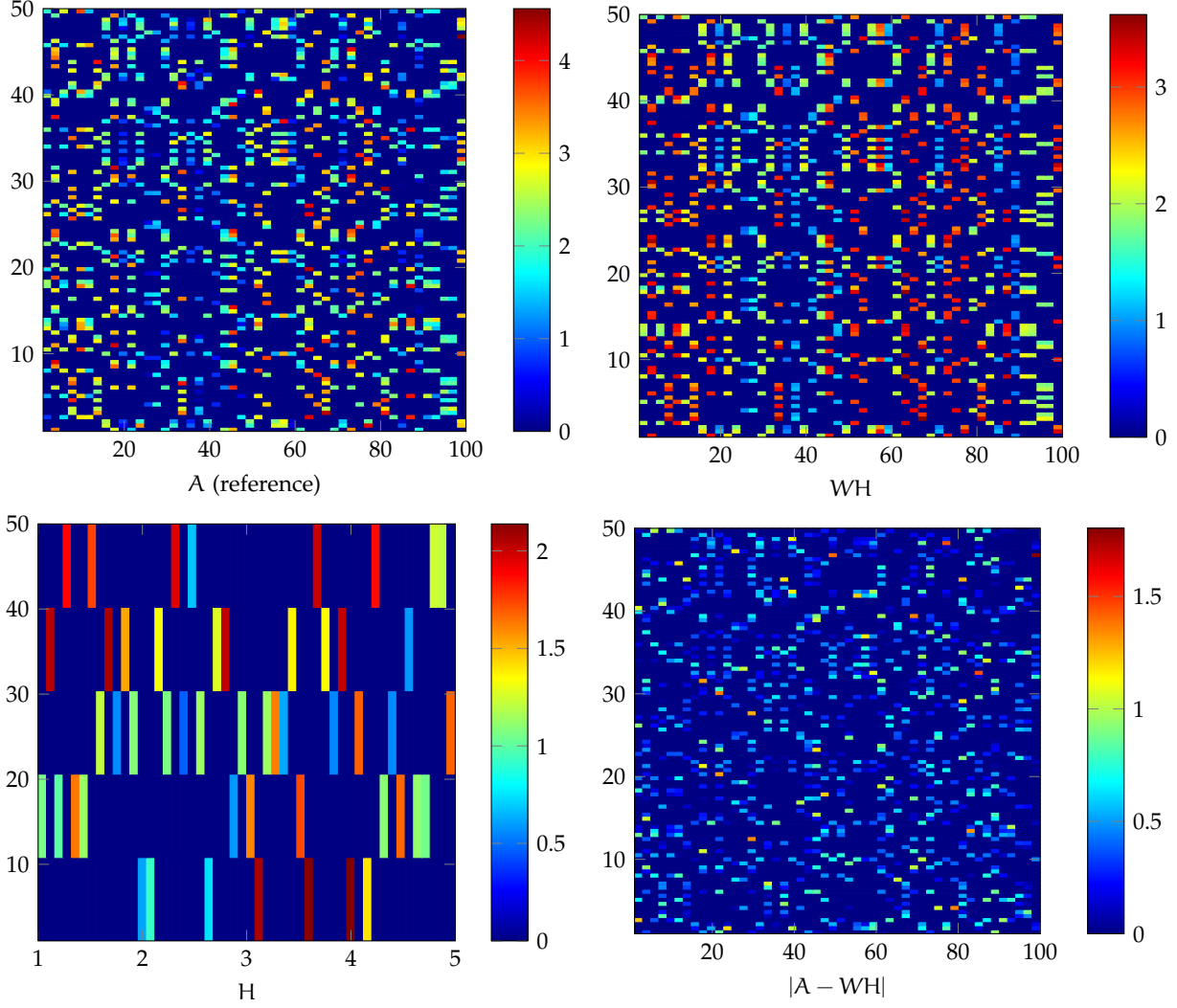


Figure 4.5 Solution of (6.84) with TR-TRDH-PSB.

which is consistent with the fact that the images show the digits sideways. The statistics are reported in Table 4.4, and show that TRDH-PSB, iTRDH-PSB, TRDH-Andrei and iTRDH-Andrei as main solvers exceed the maximum number of iterations. R2, TRDH-Spec and iTRDH-Spec are the most efficient and perform similar number of objective and gradient evaluations, with R2 terminating with a slightly higher criticality measure. TR-TRDH-PSB, TR-iTRDH-Andrei and TR-TRDH-Spec perform fewer objective and gradient evaluations, but more proximal operator evaluations than TR-R2.

Table 4.4 SVM (4.39) statistics with $h = \lambda \|\cdot\|_1$. All variants of TR use an LBFGS Hessian approximation, and are given a maximum of 100 inner iterations. The train and test accuracies are the percentage of images correctly classified, computed by counting the number of elements of the residual $\mathbf{1} - \tanh(b \odot (A^T x))$ that are lower than 1 for the train and test problems respectively.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	(Train, Test)	$\#f$	$\#\nabla f$	$\# \text{prox}$	t (s)
R2	2.18e+02	2.4e+03	1.2e-01	(99.3, 98.8)	265	199	265	3.7e+00
TRDH-Spec	2.18e+02	2.4e+03	5.9e-02	(99.3, 98.9)	306	191	611	4.4e+00
iTRDH-Spec	2.18e+02	2.4e+03	5.9e-02	(99.3, 98.9)	307	191	306	3.9e+00
TRDH-PSB	2.38e+02	2.4e+03	4.5e-01	(99.1, 98.8)	1001	686	2000	1.3e+01
iTRDH-PSB	2.38e+02	2.4e+03	4.3e-01	(99.1, 98.8)	1001	686	1000	1.3e+01
TRDH-Andrei	3.00e+02	2.9e+03	1.2e+00	(99.1, 99.0)	1001	418	2000	9.1e+00
iTRDH-Andrei	3.00e+02	2.9e+03	2.7e+00	(99.1, 99.0)	1001	418	1000	8.8e+00
TR-R2	2.18e+02	2.4e+03	4.7e-03	(99.3, 98.8)	361	361	1512	5.3e+00
TR-TRDH-PSB	2.18e+02	2.4e+03	4.7e-03	(99.3, 98.8)	316	316	3882	6.5e+00
TR-iTRDH-PSB	2.18e+02	2.4e+03	4.7e-03	(99.3, 98.8)	377	377	2242	7.8e+00
TR-TRDH-Andrei	2.18e+02	2.4e+03	4.7e-03	(99.3, 98.9)	396	396	27155	1.4e+01
TR-iTRDH-Andrei	2.18e+02	2.4e+03	4.5e-03	(99.3, 98.9)	283	283	11418	1.2e+01
TR-TRDH-Spec	2.18e+02	2.4e+03	4.7e-03	(99.3, 98.8)	333	333	3390	7.5e+00
TR-iTRDH-Spec	2.18e+02	2.4e+03	4.7e-03	(99.3, 98.8)	472	472	1791	9.3e+00

4.6.4 FitzHugh-Nagumo inverse problem

Let $v(x) = (v_1(x), \dots, v_{n+1}(x))$ and $w(x) = (w_1(x), \dots, w_{n+1}(x))$ be sampled values of $V(t; x)$ and $W(t; x)$ for $x \in \mathbb{R}^5$ satisfying the FitzHugh [42] and Nagumo *et al.* [64] model for neuron activation

$$\frac{dV}{dt} = (V - V^3/3 - W + x_1)x_2^{-1}, \quad \frac{dW}{dt} = x_2(x_3V - x_4W + x_5). \quad (4.40)$$

The samples are generated using a discretization of the time interval $t \in [0, 20]$ with initial conditions $(V(0), W(0)) = (2, 0)$. We define a target solution that corresponds to a simulation of the van der Pol [80] oscillator by generating solutions $(\bar{v}(x), \bar{w}(x))$ of (6.85) with $\bar{x} = (0, 0.2, 1, 0, 0)$, and we solve

$$\underset{x}{\text{minimize}} \frac{1}{2} \|(v(x) - \bar{v}(\bar{x}), w(x) - \bar{w}(\bar{x}))\|_2^2 + h(x), \quad (4.41)$$

where $h(x) = \lambda \|x\|_0$ with $n = 100$. The stopping tolerances ϵ_a and ϵ_r are set to 10^{-4} . We report the statistics of the solution of (4.41) with $\lambda = 10$ in Table 4.5. TRDH as main solver results in many objective and gradient evaluations compared to TR. Moreover, TR-TRDH-PSB, TR-iTRDH-PSB, TR-TRDH-Andrei and TR-iTRDH-Andrei perform fewer objective and gradient evaluations than TR-R2. TR-TRDH-Spec is the least efficient.

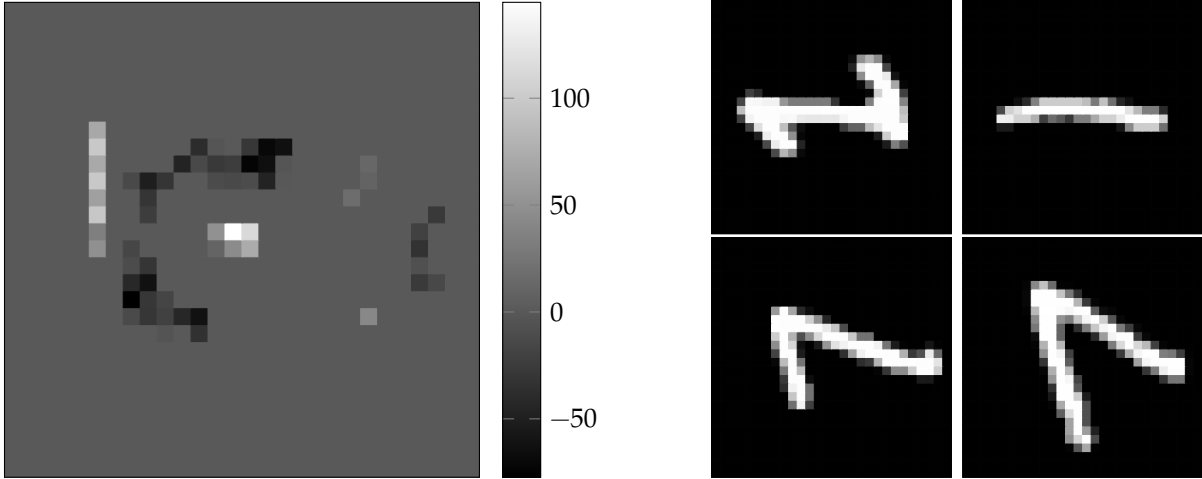


Figure 4.6 Solution map of (4.39) with TR-TRDH-Spec (left), and sample 1's (right, top row) and 7's (right, bottom row) rotated digits from the MNIST dataset.

The left half of Table 4.6 reports the final solution identified by each solver. The solution of (4.41) with TR-TRDH-PSB and the sampled values of V and W are displayed in Figure 4.7. We do not show results with R2 because it encountered numerical issues when solving the problem.

We also solve the constrained variant

$$\underset{x}{\text{minimize}} \frac{1}{2} \|(v(x) - \bar{v}(\bar{x}), w(x) - \bar{w}(\bar{x}))\|_2^2 + h(x), \quad \text{subject to } x_2 \geq 0.5, \quad (4.42)$$

and we keep all parameters the same, except for λ that we set to 40 to further enforce sparsity.

The right half of Table 4.6 reports the solution and Table 6.3 the statistics of the solve of (6.86) with the tested solvers. Note that all solvers identify the same solution. We observe that all TR-TRDH and TR-iTRDH solvers perform fewer objective and gradient evaluations than TR-R2. However, TR-R2 has the lowest number of proximal operator calls out of all solvers using TR. The right plot of Figure 4.7 shows that the W part of the solution of (6.86) with TR-TRDH-PSB does not match the data as tightly as that of (4.41) in the left half of Figure 4.7. That is a consequence of enforcing $x_2 \geq 0.5$, since the unconstrained solution verifies $x_2 = 0.2$.

4.7 Discussion and future work

The ℓ_0 - and ℓ_1 -norm regularizers are standard choices to promote sparsity. It is possible to derive the iprox of other useful separable regularizers, including ℓ_p pseudonorms to the p -th

Table 4.5 FH (4.41) statistics with $h = \lambda \|\cdot\|_0$. All variants of TR use an LBFGS Hessian approximation, and are given a maximum of 200 inner iterations. The optimal objective value is $f(x_\star) = 1.03e + 00$.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\ x - x_\star\ _2$	$\#f$	$\#\nabla f$	$\#\text{prox}$	t (s)
TRDH-Spec	3.21e+01	5	7.0e+00	9.8e−01	501	351	1000	2.1e+00
iTRDH-Spec	3.21e+01	5	5.9e+00	9.8e−01	501	351	500	2.5e+00
TRDH-PSB	1.12e+00	2	2.7e+00	1.3e−01	501	401	1000	4.7e+00
iTRDH-PSB	1.12e+00	2	2.4e+00	1.3e−01	501	401	500	4.7e+00
TRDH-Andrei	1.15e+00	2	1.6e+00	1.5e−01	501	372	1000	4.2e+00
iTRDH-Andrei	1.15e+00	2	2.8e+00	1.5e−01	501	372	500	5.3e+00
TR-R2	1.02e+00	2	5.0e−03	6.3e−03	327	236	36274	6.7e+00
TR-TRDH-PSB	1.02e+00	2	5.7e−03	6.1e−03	183	149	56042	3.8e+00
TR-iTRDH-PSB	1.02e+00	2	6.3e−03	6.2e−03	211	179	33452	3.7e+00
TR-TRDH-Andrei	1.02e+00	2	5.6e−03	6.5e−03	185	138	49311	2.9e+00
TR-iTRDH-Andrei	1.02e+00	2	6.4e−03	6.1e−03	168	151	23655	2.9e+00
TR-TRDH-Spec	1.04e+00	2	1.0e+00	7.3e−02	501	428	109750	9.0e+00
TR-iTRDH-Spec	1.02e+00	2	4.6e−03	1.1e−02	383	337	40354	6.1e+00

power, i.e.,

$$h(x) = \|x\|_p^p = \sum_{i=1}^n |x_i|^p \quad (0 < p < 1),$$

and those are also useful to promote sparsity [23]. Much remains to be done, however, including deriving the iprox of relevant non-separable regularizers, and studying other diagonal Hessian approximations than the ones considered above, including new diagonal quasi-Cauchy updates.

TRDH performs well on the problems tested and is promising, but we were surprised to see the spectral gradient update often perform better than more sophisticated diagonal quasi-Newton updates, especially when using TRDH as main solver. Future research should seek to provide an explanation for that observation. In several instances, the variant of TRDH denoted “iTRDH” performs better than the basic version, which indicates that both algorithms are relevant.

Finally, other solvers would likely benefit from using TRDH as a subproblem solver, including the methods for least-squares f of Aravkin *et al.* [2].

Acknowledgements

The authors thank Joshua Wolff from École Normale Supérieure des Techniques Avancées (ENSTA), Paris, for the work he conducted during his undergraduate internship at GERAD in the summer of 2022 and that made this research possible.

Table 4.6 FH (4.41) (left) and constrained FH (6.86) (right) solutions identified by the solvers tested. The unconstrained solution is $(0, 0.2, 1, 0, 0)$.

	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5
TRDH-Spec	0.32	0.43	0.83	0.77	0.44	0.00	0.50	0.54	0.00	0.00
iTRDH-Spec	0.32	0.43	0.83	0.77	0.44	0.00	0.50	0.54	0.00	0.00
TRDH-PSB	0.00	0.24	0.87	0.00	0.00	0.00	0.50	0.54	0.00	0.00
iTRDH-PSB	0.00	0.24	0.87	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TRDH-Andrei	0.00	0.25	0.86	0.00	0.00	0.00	0.50	0.54	0.00	0.00
iTRDH-Andrei	0.00	0.25	0.86	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-R2	0.00	0.20	1.01	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-TRDH-PSB	0.00	0.20	1.01	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-iTRDH-PSB	0.00	0.20	1.01	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-TRDH-Andrei	0.00	0.20	1.01	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-iTRDH-Andrei	0.00	0.20	1.01	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-TRDH-Spec	0.00	0.18	1.07	0.00	0.00	0.00	0.50	0.54	0.00	0.00
TR-iTRDH-Spec	0.00	0.20	1.01	0.00	0.00	0.00	0.50	0.54	0.00	0.00

Competing interests

We certify that the research submitted here is original, is our own, and is not being evaluated elsewhere for publication. This work was supported by an NSERC Discovery grant.

Data availability

The data used to produce the numerical results is available from <https://github.com/JuliaSmoothOptimizers/RegularizedProblems.jl>. The solvers are available from <https://github.com/JuliaSmoothOptimizers/RegularizedOptimization.jl> and <https://github.com/JuliaSmoothOptimizers/ShiftedProximalOperators.jl>.

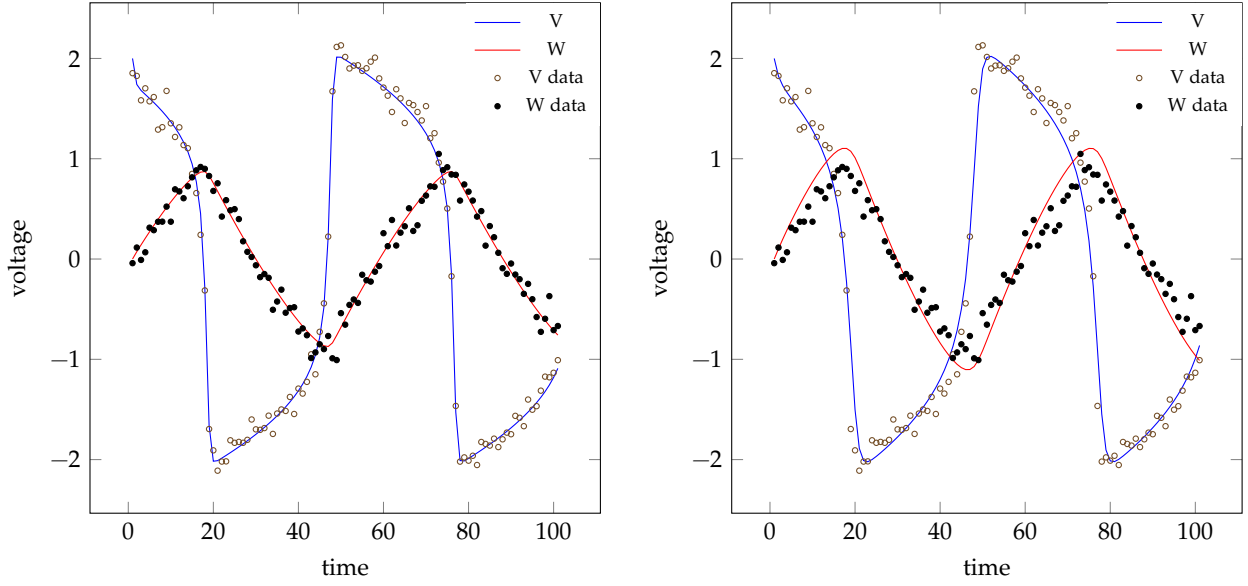


Figure 4.7 Solution of (4.41) (left) and of (6.86) (right) with TR-TRDH-PSB and sampled values of V and W .

Table 4.7 Constrained FH (6.86) statistics with $h = \lambda \|\cdot\|_1$. All variants of TR use an LBFGS Hessian approximation, and are given a maximum of 200 inner iterations.

solver	$f(x)$	$h(x)/\lambda$	ξ	$\ x - x_\star\ _2$	$\#f$	$\#\nabla f$	$\#\text{prox}$	t (s)
TRDH-Spec	4.43e+00	1.0e+00	1.2e-02	5.5e-01	327	226	653	2.6e+00
iTRDH-Spec	4.43e+00	1.0e+00	1.1e-02	5.5e-01	328	226	327	2.4e+00
TRDH-PSB	4.43e+00	1.0e+00	8.2e-03	5.5e-01	331	237	661	2.6e+00
iTRDH-PSB	4.43e+00	1.0e+00	7.7e-03	5.5e-01	332	237	331	2.2e+00
TRDH-Andrei	4.43e+00	1.0e+00	7.1e-03	5.5e-01	223	155	445	1.4e+00
iTRDH-Andrei	4.43e+00	1.0e+00	7.1e-03	5.5e-01	224	155	223	1.6e+00
TR-R2	4.43e+00	1.0e+00	4.9e-03	5.5e-01	47	31	3057	4.4e-01
TR-TRDH-PSB	4.43e+00	1.0e+00	2.7e-04	5.5e-01	38	29	6725	3.1e-01
TR-iTRDH-PSB	4.43e+00	1.0e+00	7.9e-05	5.5e-01	36	27	3326	3.3e-01
TR-TRDH-Andrei	4.43e+00	1.0e+00	8.7e-05	5.5e-01	39	30	7438	3.8e-01
TR-iTRDH-Andrei	4.43e+00	1.0e+00	1.6e-03	5.5e-01	36	27	3117	3.4e-01
TR-TRDH-Spec	4.43e+00	1.0e+00	4.1e-03	5.5e-01	36	27	5467	3.9e-01
TR-iTRDH-Spec	4.43e+00	1.0e+00	1.8e-03	5.5e-01	38	25	3347	2.5e-01

**CHAPITRE 5 ARTICLE 2: COMPLEXITY OF TRUST-REGION
METHODS WITH UNBOUNDED HESSIAN APPROXIMATIONS FOR
SMOOTH AND NONSMOOTH OPTIMIZATION**

Pour résoudre (2.72), l'algorithme 2.5.1 (TR) [4] utilise un modèle m tel que défini dans (2.73) en faisant l'hypothèse que la suite des approximations hessiennes $\{B_k\}$ est bornée. Dans ce cas, TR met $O(\epsilon^{-2})$ itérations pour que son critère d'arrêt soit satisfait avec une précision ϵ . Cependant, pour plusieurs types d'approximations quasi-Newton utilisant les équations sécantes telles que les méthodes BFGS et SR1, il n'est pas démontré que $\{B_k\}$ est bornée. [34, §8.4] indique notamment que la norme des approximations BFGS et SR1 pourrait augmenter d'une valeur inférieure ou égale à une constante strictement positive.

Le but de ce chapitre est d'établir la convergence et d'obtenir une borne sur la complexité de TR dans le cas d'approximations hessiennes non bornées. Pour cela, nous utilisons une version légèrement modifiée de TR et des hypothèses plus générales, incluant une hypothèse permettant de contrôler la croissance de $\{B_k\}$. Nous montrons que notre borne est la meilleure possible en construisant une fonction objectif f satisfaisant nos hypothèses et pour laquelle notre borne est atteinte. Notre analyse est applicable à TRDH présenté dans le chapitre 4. Elle peut aussi être utilisée pour des problèmes lisses dans le cas particulier où $h = 0$.

Title: Complexity of trust-region methods with unbounded Hessian approximations for smooth and nonsmooth optimization

Authors: Geoffroy Leconte and Dominique Orban

Submitted to: Mathematical Programming

Submission date: 26/12/2023

Abstract: We develop a worst-case evaluation complexity bound for trust-region methods in the presence of unbounded Hessian approximations. We use the algorithm of Aravkin *et al.* [4] as a model, which is designed for nonsmooth regularized problems, but applies to unconstrained smooth problems as a special case. Our analysis assumes that the growth of the Hessian approximation is controlled by the number of successful iterations. We show that the best known complexity bound of ϵ^{-2} deteriorates to $\epsilon^{-2/(1-p)}$, where $0 \leq p < 1$ is a parameter that controls the growth of the Hessian approximation. The faster the Hessian approximation

grows, the more the bound deteriorates. We construct an objective that satisfies all of our assumptions and for which our complexity bound is attained, which establishes that our bound is sharp. Numerical experiments conducted in double precision arithmetic are consistent with the theoretical analysis.

5.1 Introduction

We consider the nonsmooth regularized problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + h(x) \quad \text{subject to } \ell \leq x \leq u, \quad (5.1)$$

where $\ell \in (\mathbb{R} \cup \{-\infty\})^n$, $u \in (\mathbb{R} \cup \{+\infty\})^n$ with $\ell \leq u$ componentwise, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable on an open set containing the feasible set $[\ell, u]$ of (5.1), and $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper and lower semicontinuous (lsc). A component $\ell_i = -\infty$ or $u_i = +\infty$ indicates that x_i is unbounded below or above, respectively. Both f and h may be nonconvex. The nonsmooth regularizer h is often used to identify a local minimizer of f with desirable features, such as sparsity.

Algorithms used to solve (5.1) are often based on the proximal-gradient method [44, 61]. The trust-region method (TR) of Aravkin *et al.* [4] improves upon the proximal-gradient method by constructing a model of f and a model of h at each iteration in order to compute a step, in the spirit of traditional trust-region methods [34]. However, it was developed under the assumption that the Hessian approximations remain bounded, a common, but sometimes restrictive, assumption. A worst-case evaluation complexity bound for a stationarity measure to drop below $\epsilon \in (0, 1)$ of $O(\epsilon^{-2})$ results. The same bound holds for the algorithm of Baraldi and Kouri [7], which uses inexact function and gradient evaluations. Both match the best possible complexity bound in the smooth case, i.e., when $h = 0$ [30].

In the present paper, we examine the situation where the sequence $\{B_k\}$ of Hessian approximations is allowed to grow unbounded. In particular, it is not known whether secant approximations, such as BFGS or SR1 and the limited-memory counterparts, remain bounded. Conn *et al.* [34, §8.4] suggest that for the BFGS and SR1 approximations, the Hessian approximation could potentially grow by at most a constant at each update, though it is not clear whether that bound is attained. Aravkin *et al.* [2], Carter [24] and Lotfi *et al.* [62] present safeguarding strategies that ensure boundedness of quasi-Newton approximations. Our approach is to select an appropriate step length in the trust-region algorithm of [4] to handle unbounded approximations, whose growth we assume to be controlled by the number of *successful* iterations, i.e., those where the step is accepted by the trust-region mechanism. We develop a worst-case bound by following the techniques of Cartis *et al.* [30] in Section 5.3. We

show in Section 5.4 that the worst-case bound is indeed attained, by performing an analysis similar to that of [30, Theorem 2.2.3]. In Section 5.5, we construct an explicit function that attains the bound and validate our findings numerically.

A Julia implementation of TR is available as part of the `RegularizedOptimization.jl` package [10]. Our findings also apply to Algorithm TRDH of Leconte and Orban [54], which is similar to TR, but uses diagonal quasi-Newton approximations to compute a step without recourse to a subproblem solver.

Related research

In the smooth case where $h = 0$, under the assumption that f is convex and that there exists a constant $\kappa_{\text{uffh}} > 0$ such that for all $x \in \mathbb{R}^n$, $\|\nabla^2 f(x)\| \leq \kappa_{\text{uffh}}$, Conn *et al.* [34, §8.4] indicate that the BFGS update satisfies $\|B_{k+1}\| \leq \|B_k\| + \kappa_{\text{uffh}}$. The SR1 update satisfies a similar inequality without the convexity assumption. By assuming that

$$\sum_{k=0}^{\infty} \frac{1}{1 + \max_{0 \leq j \leq k} \|B_j\|} = \infty, \quad (5.2)$$

Toint [79] shows that the trust-region algorithm converges to a first-order stationary point.

Carter [24] presents procedures to safeguard Hessian approximations in trust-region algorithms for smooth problems. The goal of these procedures is to satisfy the *uniform predicted decrease condition*

$$\varphi_k(x_k) - \varphi_k(x_{k+1}) \geq \frac{1}{2}\beta_1 \|\nabla f(x_k)\| \min \left(\Delta_k, \frac{\|\nabla f(x_k)\|}{\beta_0} \right),$$

where β_0 and β_1 are two positive constants. When $\|B_k\| \leq \beta_0$ for all k , this condition is satisfied, but the author shows that it can also be satisfied under milder assumptions. Carter's procedures are used to correct B_k so that such assumptions hold.

Cartis *et al.* [25] show that the steepest descent method and Newton's method for smooth problems may converge in as many as $O(\epsilon^{-2})$ iterations, and that the bound is sharp for the steepest descent method. In addition, they prove that it is possible to construct an example where Newton's method is arbitrarily slow when allowing unbounded Hessians. Similarly, we show in this paper that TR may converge in as many as $O(\epsilon^{-2/(1-p)})$ iterations, where $p \in [0, 1)$ is a parameter that controls the growth of the Hessian approximations—the larger p , the larger the allowed growth. Because $\epsilon^{-2/(1-p)} \rightarrow +\infty$ as $p \nearrow 1$, our results reinforces that of Cartis *et al.* [25] and makes it more precise. The Adaptive Regularization with Cubics algorithm (ARC, or AR2 because it uses second-order derivatives) minimizes at each iteration the model

with bounded Hessian (or bounded Hessian approximations, see for instance [27, 41])

$$\varphi_k(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s + \frac{1}{3} \sigma_k \|s\|^3. \quad (5.3)$$

It is known to require at most $O(\epsilon^{-3/2})$ iterations to reach $\|\nabla f(x_k)\| \leq \epsilon$ [27, 67], and the authors show that this bound is also sharp.

The analysis of [30, Section 2.2] shows that the steepest-descent algorithm with backtracking Armijo linesearch technique results in an $O(\epsilon^{-2})$ complexity bound, and a function is constructed by polynomial interpolation to prove that the bound is sharp, with a technique that is different to that of [25].

Cartis *et al.* [29] show that Algorithm ARp for smooth problems, which is a generalization of ARC using a model of order $p \geq 1$, requires at most $O(\epsilon^{-\frac{p+1}{p}})$ iterations to satisfy $\|\nabla f(x_k)\| \leq \epsilon$. More generally, they introduce a generalization of the first-order stationarity measure $\|\nabla f(x_k)\| \leq \epsilon$ to q -th order stationarity, where $q \in \mathbb{N}_0$, and show that at most $O(\epsilon^{-\frac{p+1}{p-q+1}})$ evaluations of the objective and the derivatives are required with this measure. They require that the p -th derivative of f be globally Hölder continuous, i.e., for all x and y , $\|\nabla^p f(x) - \nabla^p f(y)\|_{[p]} \leq L\|x - y\|^\beta$, where $L \geq 0$, $\beta \in (0, 1]$, and for a symmetric tensor S where $S[v]^p$ denotes the application of S to p copies of v , $\|S\|_{[p]} = \max_{\|v\|=1} |S[v]^p|$. In addition, they construct a function by polynomial interpolation that takes exactly $\lceil \epsilon^{-\frac{p+1}{p-q+1}} \rceil$ iterations to satisfy the criticality measure, which indicates that the complexity bound is sharp.

Cartis *et al.* [26] describe a first-order trust-region method and a quadratic regularization algorithm to solve nonsmooth problem of the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + h(c(x)), \quad (5.4)$$

where f and c are continuously differentiable and may be nonconvex, and h is convex but may be nonsmooth. Note that (5.4) is a special case of (5.1). They show that both algorithms have a complexity bound of $O(\epsilon^{-2})$.

Aravkin *et al.* [4] describe a proximal trust-region algorithm to solve (5.1) using bounded quasi-Newton approximations. They also present a quadratic regularization variant. They establish that their criticality measure is smaller than ϵ in at most $O(\epsilon^{-2})$ iterations for both algorithms. Aravkin *et al.* [2] adapt these algorithms to solve nonsmooth regularized least-squares problems and obtain the same complexity bound.

Baraldi and Kouri [7] also describe a proximal trust-region algorithm for convex regularizers

h . In addition, they allow the use of inexact objective and gradient evaluations. As Toint [79] in the smooth case, they assume that

$$\sum_{k=0}^{\infty} \frac{1}{1 + \max_{0 \leq j \leq k} \omega_j} = \infty, \quad (5.5)$$

where

$$\omega_k = \sup \left\{ \frac{2}{\|s\|^2} |\varphi_k(x_k + s) - \varphi_k(x_k) - \nabla \varphi_k(x_k)^T s| \mid 0 < \|s\| \leq \Delta_k \right\},$$

and φ_k is a smooth model of f about x_k . In particular, if φ_k is a second-order Taylor approximation at x_k with Hessian approximation B_k , $\omega_k = \sup \{s^T B_k s / \|s\|^2 \mid 0 < \|s\| \leq \Delta_k\}$, so that (5.5) is reminiscent of (5.2). If ω_k is bounded independently of k , which is the case for bounded Hessian approximations, they show that their algorithm enjoys a complexity bound of $O(\epsilon^{-2})$.

Cartis *et al.* [28] present a similar concept of high-order approximate minimizers to that of [29] for nonsmooth problems such as (5.4) where f, c are smooth, and h is nonsmooth and Lipschitz-continuous. They present an algorithm of adaptive regularization of order p , and derive several bounds depending on the properties of (5.4) and of the order of the desired approximate minimizer. In particular, for the order $q = 1$ and convex regularizer h , their complexity bound is $O(\epsilon^{-\frac{p+1}{p}})$, and they show that it is sharp by constructing a function by polynomial interpolation.

Notation

$\mathbb{B}$ denotes the unit ball at the origin in a certain norm dictated by the context, $\Delta \mathbb{B}$ is the ball of radius $\Delta > 0$ centered at the origin, and $x + \Delta \mathbb{B}$ is the ball of radius $\Delta > 0$ centered at $x \in \mathbb{R}^n$. For $A \subseteq \mathbb{R}^n$, the indicator of A is $\chi(\cdot \mid A) : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ defined as $\chi(x \mid A) = 0$ if $x \in A$ and $+\infty$ otherwise. If $A \neq \emptyset$, $\chi(\cdot \mid A)$ is proper. If A is closed, $\chi(\cdot \mid A)$ is lsc. For a finite set $A \subset \mathbb{N}$, we denote $|A|$ its cardinality. If f_1 and f_2 are two positive functions of $\epsilon > 0$, we say that $f_1(\epsilon) = O(f_2(\epsilon))$ if there exists a constant $C > 0$ such that $f_1(\epsilon) \leq C f_2(\epsilon)$ for all $\epsilon > 0$ sufficiently small.

5.2 Context

5.2.1 Background

We recall relevant concepts of variational analysis—see, e.g., [73].

Consider $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $\bar{x} \in \mathbb{R}^n$ with $\phi(\bar{x}) < \infty$. The *Fréchet subdifferential* of ϕ at $\bar{x}$ is the

closed convex set $\widehat{\partial}\phi(\bar{x})$ of $v \in \mathbb{R}^n$ such that

$$\liminf_{\substack{x \rightarrow \bar{x} \\ x \neq \bar{x}}} \frac{\phi(x) - \phi(\bar{x}) - v^T(x - \bar{x})}{\|x - \bar{x}\|} \geq 0.$$

The *limiting subdifferential* of ϕ at $\bar{x}$ is the closed, but not necessarily convex, set $\partial\phi(\bar{x})$ of $v \in \mathbb{R}^n$ for which there exist $\{x_k\} \rightarrow \bar{x}$ and $\{v_k\} \rightarrow v$ such that $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$ and $v_k \in \widehat{\partial}\phi(x_k)$ for all k .

We say that $\bar{x}$ is *stationary* for the problem of minimizing ϕ if $0 \in \partial\phi(\bar{x})$.

The *horizon subdifferential* of ϕ at $\bar{x}$ is the closed, but not necessarily convex, cone $\partial^\infty\phi(\bar{x})$ of $v \in \mathbb{R}^n$ for which there exist $\{x_k\} \rightarrow \bar{x}$, $\{v_k\}$ and $\{\lambda_k\} \downarrow 0$ such that $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$, $v_k \in \widehat{\partial}\phi(x_k)$ for all k , and $\{\lambda_k v_k\} \rightarrow v$. $\widehat{\partial}\phi(\bar{x}) \subset \partial\phi(\bar{x})$ always holds.

If $C \subseteq \mathbb{R}^n$ and $\bar{x} \in C$, the closed convex cone $\widehat{N}_C(\bar{x}) := \widehat{\partial}\chi(\bar{x} \mid C)$ is the regular normal cone to C at $\bar{x}$. The closed cone $N_C(\bar{x}) := \partial\chi(\bar{x} \mid C) = \partial^\infty\chi(\bar{x} \mid C)$ is the normal cone to C at $\bar{x}$. $\widehat{N}_C(\bar{x}) \subseteq N_C(\bar{x})$ always holds, and is an equality if C is convex.

ϕ is *proper* if $\phi(x) > -\infty$ for all x , and $\phi(x) < \infty$ for at least one x . ϕ is *lower semicontinuous* (*lsc*) at $\bar{x}$ if $\liminf_{x \rightarrow \bar{x}} \phi(x) = \phi(\bar{x})$.

Let $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ be proper lsc, and $C \subseteq \mathbb{R}^n$ be closed. We say that the *constraint qualification* is satisfied at $\bar{x} \in C$ for the constrained problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \phi(x) \quad \text{subject to } x \in C \tag{5.6}$$

if

$$\partial^\infty\phi(\bar{x}) \cap N_C(\bar{x}) = \{0\}. \tag{5.7}$$

If $\bar{x}$ solves (5.6) and (5.7) is satisfied at $\bar{x}$, [73, Theorem 8.15] yields

$$0 \in \partial(\phi + \chi(\cdot \mid C))(\bar{x}) = \partial\phi(\bar{x}) + N_C(\bar{x}).$$

In the case of (5.1), this first-order necessary condition for optimality reads

$$0 \in \nabla f(\bar{x}) + \partial h(\bar{x}) + N_{[\ell, u]}(\bar{x})$$

thanks to [73, Exercise 8.8c].

The *proximal operator* associated with a proper lsc function ϕ is

$$\text{prox}_{\nu\phi}(q) := \underset{x}{\operatorname{argmin}} \quad \frac{1}{2}\nu^{-1}\|x - q\|_2^2 + \phi(x), \quad (5.8)$$

where $\nu > 0$ is a preset steplength.

If ϕ is prox-bounded and $\nu > 0$ is sufficiently small, $\text{prox}_{\nu\phi}(q)$ is a nonempty and closed set. It may contain multiple elements.

The proximal gradient method [44, 61] for (5.1) is a generalization of the gradient method that takes the nonsmooth term into account. It generates iterates $\{s_j\}$ according to

$$s_{j+1} \in \underset{\nu h}{\operatorname{prox}}(s_j - \nu \nabla f(s_j)). \quad (5.9)$$

5.2.2 Models and trust-region algorithm

At $x \in \mathbb{R}^n$ where h is finite, we define models

$$\varphi(s; x) \approx f(x + s) \quad (5.10a)$$

$$\psi(s; x) \approx h(x + s) \quad (5.10b)$$

$$m(s; x) := \varphi(s; x) + \psi(s; x), \quad (5.10c)$$

Our assumptions on (5.10) are the same as those of Aravkin *et al.* [4]:

Model Assumption 5.2.1. *For any $x \in \mathbb{R}^n$, $\varphi(\cdot; x) \in \mathcal{C}^1$, and satisfies $\varphi(0; x) = f(x)$ and $\nabla \varphi(0; x) = \nabla f(x)$. For any $x \in \mathbb{R}^n$ where h is finite, ψ is proper lsc, and satisfies $\psi(0; x) = h(x)$ and $\partial \psi(0) = \partial h(x)$.*

The following result states that if $s = 0$ minimizes (5.10c) and (5.7) is satisfied, x must be stationary.

Proposition 13 (54, Proposition 1). *Let $C \subset \mathbb{R}^n$ be nonempty and compact, and let Model Assumption 5.2.1 be satisfied. Let (5.1) satisfy the constraint qualification (5.7) at $x \in C$. Assume $0 \in \operatorname{argmin}_s m(s; x) + \chi(x + s \mid C)$, and let the latter subproblem satisfy the constraint qualification (5.7) at $s = 0$. Then x is first-order stationary for (5.1).*

A useful model is based on the second-order Taylor expansion

$$\varphi(s; x, B) := f(x) + \nabla f(x)^T s + \frac{1}{2} s^T B s, \quad (5.11a)$$

$$m(s; x, B) := \varphi(s; x, B) + \psi(s; x), \quad (5.11b)$$

where $B = B^T \in \mathbb{R}^{n \times n}$.

Each iteration is divided into two parts. In the first part, Aravkin *et al.* [3] define the following model based on a first-order Taylor expansion to compute a *Cauchy point*

$$\varphi_{\text{cp}}(s; x) := f(x) + \nabla f(x)^T s, \quad (5.12a)$$

$$m(s; x, \nu) := \varphi_{\text{cp}}(s; x) + \frac{1}{2}\nu^{-1}\|s\|^2 + \psi(s; x), \quad (5.12b)$$

where $\nu_k > 0$ and “cp” stands for “Cauchy point.” We compute a first step

$$s_{k,1} \in \underset{s}{\operatorname{argmin}} \quad m(s; x_k, \nu_k) + \chi(x_k + s \mid [\ell, u] \cap (x_k + \Delta_k \mathbb{B})), \quad (5.13)$$

for an appropriate value of $\nu_k > 0$.

In the notation of [3], let

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) := f(x_k) + h(x_k) - \varphi_{\text{cp}}(s_{k,1}; x_k) - \psi(s_{k,1}; x_k), \quad (5.14)$$

denote the optimal model decrease for (5.12). The following proposition indicates that $\xi_{\text{cp}}(\Delta; x, \nu)$ can be used to determine whether x is first-order stationary for (5.1).

Proposition 14 (4, Proposition 3.3 and 3). *Let Model Assumption 5.2.1 be satisfied, $\Delta > 0$, and $\nu > 0$. In addition, let (5.1) satisfy the constraint qualification at x and (5.12) satisfy the constraint qualification at 0. Then, $\xi_{\text{cp}}(\Delta; x, \nu) = 0 \iff 0$ is a solution of (5.13) $\implies x$ is first-order stationary for (5.1).*

In the second part of iteration k , we construct $m_k(s; x_k, B_k) := \varphi(s; x_k, B_k) + \psi(s; x_k) \approx f(x_k + s) + h(x_k + s)$, and compute an approximate solution of

$$\underset{s}{\operatorname{minimize}} \quad m_k(s; x_k) \quad \text{subject to } \|s\| \leq \Delta_k, \quad (5.15)$$

using $s_{k,1}$ as starting point.

We focus on the trust-region (TR) algorithm formally stated as Algorithm 5.2.1. It consists of the algorithm of Aravkin *et al.* [4] with a modified maximum allowable stepsize ν_k .

In Algorithm 5.2.1, $s_{k,1}$ is used to check for stationarity, and to set the trust-region radius for the computation of the search direction s_k .

Let us now briefly turn our attention to unconstrained smooth problems. In this case, the following lemma gives a global minimizer of (5.12) and (5.11).

Algorithm 5.2.1 Nonsmooth trust-region algorithm with potentially unbounded Hessian.

1: Choose constants

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3 \leq \gamma_4, \quad \Delta_{\max} > \Delta_0, \quad \alpha > 0, \quad \text{and} \quad \beta \geq 1.$$

2: Choose a stopping tolerance $\epsilon > 0$.

3: Choose $x_0 \in \mathbb{R}^n$ where h is finite, $\Delta_0 > 0$, compute $f(x_0) + h(x_0)$.

4: **for** $k = 0, 1, \dots$ **do**

5: Choose

$$0 < \nu_k \leq \frac{\alpha \Delta_k}{1 + \|B_k\|(1 + \alpha \Delta_k)} = \frac{1}{\alpha^{-1} \Delta_k^{-1} + \|B_k\|(1 + \alpha^{-1} \Delta_k^{-1})}.$$

6: Define $m_k(s; x_k, \nu_k)$ as in (5.12) and compute $s_{k,1}$ as in (5.13).

7: If $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \leq \epsilon$, terminate and claim that x_k is approximately stationary.

8: Define $m_k(s; x_k, B_k)$ as in (5.11) according to Model Assumption 5.2.1 and compute a solution s_k of (5.15) with Δ_k replaced by $\min(\Delta_k, \beta \|s_{k,1}\|)$.

9: Compute the ratio

$$\rho_k := \frac{f(x_k) + h(x_k) - (f(x_k + s_k) + h(x_k + s_k))}{m_k(0; x_k, B_k) - m_k(s_k; x_k, B_k)}. \quad (5.16)$$

10: If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$. Otherwise, set $x_{k+1} = x_k$.

11: Update the trust-region radius according to

$$\bar{\Delta}_{k+1} \in \begin{cases} [\gamma_3 \Delta_k, \gamma_4 \Delta_k] & \text{if } \rho_k \geq \eta_2, & \text{(very successful iteration)} \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \eta_1 \leq \rho_k < \eta_2, & \text{(successful iteration)} \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1, & \text{(unsuccessful iteration)} \end{cases}$$

$$\text{and } \Delta_{k+1} = \min(\bar{\Delta}_{k+1}, \Delta_{\max})$$

Lemma 4. *We consider the special case of (5.1) where $h = 0$, $\ell_i = -\infty$ and $u_i = +\infty$ for $i = 1, \dots, n$. Let $B = B^T \in \mathbb{R}^{n \times n}$ be positive definite and $\psi = 0$. Then for any $x \in \mathbb{R}^n$,*

$$\operatorname{argmin}_s m(s; x, B) = \operatorname{argmin}_s \varphi(s; x, B) = \{-B^{-1} \nabla f(x)\}. \quad (5.17)$$

In particular, if $B = \nu^{-1}I$ with $\nu > 0$,

$$\operatorname{argmin}_s m(s; x, \nu) = \operatorname{argmin}_s \varphi_{\text{cp}}(s; x) + \frac{1}{2} \nu^{-1} \|s\|^2 = \{s_{k,1}\} = \{-\nu \nabla f(x)\}. \quad (5.18)$$

Proof. The objective of (5.17) is convex because B is positive definite. Its global minimizer satisfies the first-order necessary condition $\nabla f(x) + Bs = 0$, i.e., $s = -B^{-1} \nabla f(x)$. With $B = \nu^{-1}I$, the first-order necessary condition is $s = -\nu \nabla f(x)$. $\square$

The following proposition draws a parallel between $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)$ and $\|\nabla f(x_k)\|$ for smooth problems when the trust-region constraint is inactive, as is expected to occur when close to a stationary point.

Proposition 15. *We consider the special case of (5.1) where $h = 0$, $\ell_i = -\infty$ and $u_i = +\infty$ for $i = 1, \dots, n$. If $\|s_{k,1}\| < \Delta_k$, then $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = \nu_k \|\nabla f(x_k)\|^2$.*

Proof. If the trust-region constraint is inactive, Lemma 4 indicates that $s_{k,1} = -\nu_k \nabla f(x_k)$. Thus, (5.14) yields $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = -\nabla f(x_k)^T s_{k,1} = \nu_k \|\nabla f(x_k)\|^2$. $\square$

5.3 Convergence and complexity with potentially unbounded Hessian

From this section onwards, we consider the quasi-Newton model defined in (5.11), and we aim to establish convergence and worst-case complexity results for Algorithm 5.2.1 in the presence of potentially unbounded Hessian approximations B_k .

The following two assumptions are essential. Assumption 2 is [4, Step Assumption 3.8b], whereas Assumption 3 is a relaxed version of [4, Step Assumption 3.8a] that takes into account potentially unbounded Hessian approximations. Indeed, assuming, for simplicity, that $\nabla^2 f(x_k)$ exists, a second-order Taylor expansion of f about x_k yields

$$f(x_k + s_k) - \varphi(s_k; x_k, B_k) = \frac{1}{2} s_k^T (\nabla^2 f(x_k) - B_k) s_k + o(\|s_k\|^2),$$

which is not necessarily $O(\|s_k\|^2)$ if $\{B_k\}$ is unbounded.

Assumption 2. *There exists $\kappa_{\text{mdc}} \in (0, 1)$ such that*

$$m(0; x_k, B_k) - m(s_k; x_k, B_k) \geq \kappa_{\text{mdc}} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k). \quad (5.19)$$

Assumption 3. *There exists $\kappa_{\text{ubd}} > 0$ such that*

$$|(f + h)(x_k + s_k) - m(s_k; x_k, B_k)| \leq \kappa_{\text{ubd}} (1 + \|B_k\|) \|s_k\|_2^2. \quad (5.20)$$

Leconte and Orban [54, Proposition 2] and Aravkin *et al.* [3] already indicate that Assumption 2 holds for TRDH and TR. We now justify that it also holds for Algorithm 5.2.1 with potentially unbounded Hessian approximations.

Proposition 16. *If Model Assumption 5.2.1 is satisfied, there exists $\kappa_{\text{mdc}} \in (0, 1)$ such that Assumption 2 holds.*

Proof. We proceed similarly as in [54, Proposition 2]. The definition of s_k implies that

$$m(s_k; x_k, B_k) \leq m(s_{k,1}; x_k, B_k) = \varphi_{\text{cp}}(s_{k,1}; x_k) + \frac{1}{2} s_{k,1}^T B_k s_{k,1} + \psi(s_{k,1}; x_k).$$

As

$$s_{k,1}^T B_k s_{k,1} \leq |s_{k,1}^T B_k s_k| \leq \|s_{k,1}\| \|B_k s_{k,1}\| \leq \|B_k\| \|s_{k,1}\|^2,$$

where we used Cauchy-Schwarz in the second inequality and the consistency of the ℓ_2 -norm for matrices in the third inequality,

$$m(s_k; x_k, B_k) \leq \varphi_{\text{cp}}(s_{k,1}; x_k) + \frac{1}{2} \|B_k\| \|s_{k,1}\|^2 + \psi(s_{k,1}; x_k),$$

which leads to

$$m(0; x_k, B_k) - m(s_k; x_k, B_k) \geq \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) - \frac{1}{2} \|B_k\| \|s_{k,1}\|^2.$$

To satisfy Assumption 2, it is sufficient to show that there exists $\kappa_{\text{mdc}} \in (0, 1)$ such that

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) - \frac{1}{2} \|B_k\| \|s_{k,1}\|^2 \geq \kappa_{\text{mdc}} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k),$$

i.e.,

$$(1 - \kappa_{\text{mdc}}) \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2} \|B_k\| \|s_{k,1}\|^2.$$

As $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \geq \frac{1}{2} \nu_k^{-1} \|s_{k,1}\|^2$ by definition of $s_{k,1}$ and $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)$, it is also sufficient to show that there exists $\kappa_{\text{mdc}} \in (0, 1)$ such that

$$(1 - \kappa_{\text{mdc}}) \nu_k^{-1} \geq \|B_k\|. \quad (5.21)$$

Finally,

$$\begin{aligned} \|B_k\| \nu_k &= \frac{1}{\alpha^{-1} \Delta_k^{-1} \|B_k\|^{-1} + 1 + \alpha^{-1} \Delta_k^{-1}} \leq \frac{1}{\alpha^{-1} \Delta_{\max}^{-1} \|B_k\|^{-1} + 1 + \alpha^{-1} \Delta_{\max}^{-1}} \\ &\leq \frac{1}{1 + \alpha^{-1} \Delta_{\max}^{-1}} \in (0, 1). \end{aligned} \quad (5.22)$$

We deduce from (5.22) that (5.21) holds, which is sufficient to satisfy Assumption 2. $\square$

We begin the convergence analysis by showing that there still exists a Δ_{succ} as in [4, Theorem 3.4], despite our more general Assumption 3.

Theorem 1. *Let Model Assumption 5.2.1, Assumption 2 and Assumption 3 be satisfied and*

$$\Delta_{\text{succ}} := \frac{\kappa_{\text{mdc}}(1 - \eta_2)}{2\kappa_{\text{ubd}}\alpha\beta^2} > 0.$$

If (5.1) satisfies the constraint qualification at x_k , (5.12) satisfies the constraint qualification at 0, x_k is not first-order stationary for (5.1), and $\Delta_k \leq \Delta_{\text{succ}}$, then iteration k is very successful and $\Delta_{k+1} \geq \Delta_k$.

Proof. Lemma 2 of [17] guarantees that

$$\begin{aligned} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) &\geq \frac{1}{2}\nu_k^{-1}\|s_{k,1}\|^2 \\ &\geq \frac{1}{2}(\alpha^{-1}\Delta_k^{-1} + \|B_k\|(1 + \alpha^{-1}\Delta_k^{-1}))\|s_{k,1}\|^2 \\ &\geq \frac{1}{2}(\alpha^{-1}\Delta_k^{-1}(1 + \|B_k\|))\|s_{k,1}\|^2. \end{aligned} \tag{5.23}$$

If $\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = 0$, then $s_{k,1} = 0$, and x_k is first-order stationary with Proposition 13. If x_k is not first-order stationary, $s_{k,1} \neq 0$ according to Proposition 14. In this case, Assumption 2, Assumption 3, and (5.23) lead to

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{(f + h)(x_k + s_k) - m(s_k; x_k, B_k)}{m(0; x_k, B_k) - m(s_k; x_k, B_k)} \right| \\ &\leq \frac{\kappa_{\text{ubd}}(1 + \|B_k\|)\|s_k\|_2^2}{\kappa_{\text{mdc}}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)} \\ &\leq \frac{\kappa_{\text{ubd}}(1 + \|B_k\|)\beta^2\|s_{k,1}\|_2^2}{\frac{1}{2}\kappa_{\text{mdc}}\alpha^{-1}\Delta_k^{-1}(1 + \|B_k\|)\|s_{k,1}\|^2} \\ &= \frac{2\kappa_{\text{ubd}}\beta^2\alpha\Delta_k}{\kappa_{\text{mdc}}}. \end{aligned}$$

Thus, $\Delta_k \leq \Delta_{\text{succ}}$ implies $\rho_k \geq \eta_2$ and iteration k is very successful. $\square$

We set $\Delta_{\min} := \min(\Delta_0, \gamma_1\Delta_{\text{succ}})$, and we observe that $\Delta_k \geq \Delta_{\min}$ for all $k \in \mathbb{N}$. We use $\nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ as our criticality measure. Let $0 < \epsilon < 1$, and

$$\begin{aligned} I(\epsilon) &:= \{k \in \mathbb{N} \mid \nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} > \epsilon\}, \\ S(\epsilon) &:= \{k \in I(\epsilon) \mid \rho_k \geq \eta_1\}, \\ U(\epsilon) &:= \{k \in I(\epsilon) \mid \rho_k < \eta_1\}, \end{aligned}$$

be the set of iterations, successful iterations, and unsuccessful iterations until the criticality measure drops below ϵ , respectively.

At iteration k of Algorithm 5.2.1, let σ_k be the number of successful iterations encountered so far:

$$\sigma_k = |\{j = 0, \dots, k \mid \rho_j \geq \eta_1\}|, \quad k \in \mathbb{N}. \quad (5.24)$$

We introduce an assumption allowing $\{B_k\}$ to be unbounded, as long as it is controlled by σ_k .

Assumption 4. *There are constants $\mu > 0$ and $0 \leq p < 1$ such that $\max_{0 \leq j \leq k} \|B_j\| \leq \mu(1 + \sigma_k^p)$ for all $k \in \mathbb{N}$.*

Clearly, Assumption 4 allows approximations that grow unbounded, though they must not grow too fast. It reduces to the bounded case when $p = 0$. We may now establish a variant of [4, Lemma 3.6] based on Assumption 4. When $p = 0$, Aravkin *et al.* [4] show that $|S(\epsilon)| = O(\epsilon^{-2})$. In the following result, we restrict our attention to the case $p > 0$.

Lemma 5. *Let Assumption 2 and Assumption 4 be satisfied with $p > 0$. Assume that Algorithm 5.2.1 generates infinitely many successful iterations, that the step size $\nu_k := \alpha\Delta_k/(1 + \|B_k\|(1 + \alpha\Delta_k))$ is selected at each iteration, and that there exists $(f + h)_{\text{low}} \in \mathbb{R}$ such that $(f + h)(x_k) \geq (f + h)_{\text{low}}$ for all $k \in \mathbb{N}$. Let $\epsilon \in (0, 1)$ be small enough that $\mu + 1 \leq \mu|S(\epsilon)|^p$. Then,*

$$|S(\epsilon)| \leq \left(2\mu(1 + \alpha^{-1}\Delta_{\min}^{-1}) \frac{(f + h)(x_0) - (f + h)_{\text{low}}}{\eta_1 \kappa_{\text{mdc}} \epsilon^2} \right)^{1/(1-p)} = O\left(\epsilon^{-2/(1-p)}\right). \quad (5.25)$$

Proof. Let $k \in S(\epsilon)$. We have

$$\begin{aligned} (f + h)(x_k) - (f + h)(x_k + s_k) &\geq \eta_1 \kappa_{\text{mdc}} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k) \\ &\geq \eta_1 \kappa_{\text{mdc}} \nu_k \epsilon^2 \\ &= \eta_1 \kappa_{\text{mdc}} \frac{1}{\alpha^{-1}\Delta_k^{-1} + \|B_k\|(1 + \alpha^{-1}\Delta_k^{-1})} \epsilon^2 \\ &\geq \eta_1 \kappa_{\text{mdc}} \frac{1}{\alpha^{-1}\Delta_{\min}^{-1} + \|B_k\|(1 + \alpha^{-1}\Delta_{\min}^{-1})} \epsilon^2. \end{aligned}$$

We add together the above inequalities over all $k \in S(\epsilon)$ and use the assumption that $f + h$ is

bounded below to obtain

$$\begin{aligned}
(f+h)(x_0) - (f+h)_{\text{low}} &\geq \eta_1 \kappa_{\text{mdc}} \epsilon^2 \sum_{k \in S(\epsilon)} \frac{1}{\alpha^{-1} \Delta_{\min}^{-1} + \|B_k\| (1 + \alpha^{-1} \Delta_{\min}^{-1})} \\
&\geq \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)| \min_{k \in S(\epsilon)} \frac{1}{\alpha^{-1} \Delta_{\min}^{-1} + \|B_k\| (1 + \alpha^{-1} \Delta_{\min}^{-1})} \\
&= \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)| \frac{1}{\max_{k \in S(\epsilon)} (\alpha^{-1} \Delta_{\min}^{-1} + \|B_k\| (1 + \alpha^{-1} \Delta_{\min}^{-1}))} \\
&= \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)| \frac{1}{\alpha^{-1} \Delta_{\min}^{-1} + (\max_{k \in S(\epsilon)} \|B_k\|) (1 + \alpha^{-1} \Delta_{\min}^{-1})} \\
&\geq \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)| \frac{1}{\alpha^{-1} \Delta_{\min}^{-1} + \mu (1 + |S(\epsilon)|^p) (1 + \alpha^{-1} \Delta_{\min}^{-1})} \\
&\geq \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)| \frac{1}{(\mu + 1 + \mu |S(\epsilon)|^p) (1 + \alpha^{-1} \Delta_{\min}^{-1})},
\end{aligned}$$

where we appealed to Assumption 4 in the penultimate step.

Because, $\mu + 1 \leq \mu |S(\epsilon)|^p$,

$$\begin{aligned}
(f+h)(x_0) - (f+h)_{\text{low}} &\geq \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)| \frac{1}{2\mu |S(\epsilon)|^p (1 + \alpha^{-1} \Delta_{\min}^{-1})} \\
&= \eta_1 \kappa_{\text{mdc}} \epsilon^2 |S(\epsilon)|^{1-p} \frac{1}{2\mu (1 + \alpha^{-1} \Delta_{\min}^{-1})},
\end{aligned}$$

which establishes (5.25). $\square$

If there are infinitely many successful iterations, the inequality $\mu + 1 > \mu |S(\epsilon)|^p$ can only hold for all sufficiently small $\epsilon > 0$ if $p = 0$.

The complexity bound $|S(\epsilon)| = O(\epsilon^{-2/(1-p)})$ also holds for $p = 0$, as it reduces to that of Aravkin *et al.* [4]. We obtain a complexity bound of $O(\epsilon^{-5/2})$ for $p = \frac{1}{5}$ and $O(\epsilon^{-3})$ for $p = \frac{1}{3}$. In other words, the faster the growth of $\|B_k\|$, the worse the deterioration of the complexity bound.

A bound on the number of unsuccessful iteration is obtained using the technique of Cartis *et al.* [30].

Proposition 17 (4, Lemma 3.7). *Under the assumptions of Lemma 5,*

$$|U(\epsilon)| \leq \log_{\gamma_2}(\Delta_{\min}/\Delta_0) + |S(\epsilon)| |\log_{\gamma_2}(\gamma_4)|. \quad (5.26)$$

Proof. The proof is a minor modification of that of [4, Lemma 3.7]. We provide it for completeness. Let k_ϵ be the smallest integer satisfying $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} \leq \epsilon$. The update

rule of Δ_k in Line 11 indicates that

$$\Delta_{\min} \leq \Delta_{k_\epsilon-1} \leq \min(\Delta_0 \gamma_2^{|U(\epsilon)|} \gamma_4^{|S(\epsilon)|}, \Delta_{\max}) \leq \Delta_0 \gamma_2^{|U(\epsilon)|} \gamma_4^{|S(\epsilon)|}.$$

As $0 < \gamma_2 < 1$, we take the logarithm of the above inequalities to obtain

$$|U(\epsilon)| \log(\gamma_2) + |S(\epsilon)| \log(\gamma_4) \geq \log(\Delta_{\min}/\Delta_0),$$

which leads to (5.26). $\square$

Thus, under Assumption 4, Lemma 5 and Proposition 17 show that

$$\liminf \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = 0.$$

5.4 Sharpness of the complexity bound

In this section, we show that the bound of Lemma 5 is attained using the techniques of Cartis *et al.* [30, Theorem 2.2.3]. To this end, for $0 < \epsilon \leq 1/2$, we explicitly construct $k_\epsilon = \lfloor \epsilon^{-2/(1-p)} \rfloor$ iterates of Algorithm 5.2.1 with $n = 1$ and $h = 0$, so that $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} > \epsilon$ for $k = 0, \dots, k_\epsilon - 1$, and $\nu_{k_\epsilon}^{-1/2} \xi(\Delta_{k_\epsilon}; x_{k_\epsilon}, \nu_{k_\epsilon})^{1/2} = \epsilon$. Then, we invoke [30, Theorem A.9.2] to establish that there exists $f : \mathbb{R} \rightarrow \mathbb{R}$ in (5.1) that interpolates our iterates and satisfies our assumptions. The following result is a special case of [30, Theorem A.9.2].

Proposition 18 (Hermite interpolation with function and gradient evaluations). *Let k_ϵ be a positive integer, $\{f_k\}$, $\{g_k\}$ and $\{x_k\}$ be sequences of numbers given for $k \in \{0, \dots, k_\epsilon\}$. Assume that for $k \in \{0, \dots, k_\epsilon\}$, $s_k = x_{k+1} - x_k > 0$, and that for all $k \in \{0, \dots, k_\epsilon - 1\}$,*

$$|f_{k+1} - (f_k + g_k s_k)| \leq \kappa_f s_k^2, \tag{5.27a}$$

$$|g_{k+1} - g_k| \leq \kappa_f s_k, \tag{5.27b}$$

for some constant $\kappa_f \geq 0$. Then, there exists $f : \mathbb{R} \rightarrow \mathbb{R}$ continuously differentiable such that

$$f(x_k) = f_k \quad \text{and} \quad f'(x_k) = g_k.$$

In addition, if

$$|f_k| \leq \kappa_f, \quad |g_k| \leq \kappa_f \quad \text{and} \quad s_k \leq \kappa_f,$$

then $|f|$ and $|f'|$ are bounded by a constant depending only on κ_f .

Proof. The result is a special case of [30, Theorem A.9.2] with $p = 1$. $\square$

In the following, we use

$$0 < \epsilon \leq 1/2, \quad (5.28a)$$

$$0 \leq p < 1, \quad (5.28b)$$

$$k_\epsilon = \lfloor \epsilon^{-2/(1-p)} \rfloor, \quad (5.28c)$$

$$\alpha > 0, \quad (5.28d)$$

$$\beta \geq 2\alpha^{-1} + 1, \quad (5.28e)$$

and for all $k \in \{0, \dots, k_\epsilon\}$, we define the sequences

$$w_k := (k_\epsilon - k)/k_\epsilon, \quad (5.29a)$$

$$g_k := -\epsilon(1 + w_k). \quad (5.29b)$$

In addition, using the initial values

$$\Delta_0 := 1, \quad (5.30a)$$

$$B_0 := 1, \quad (5.30b)$$

$$s_0 := -g_0, \quad (5.30c)$$

$$x_0 := 0, \quad (5.30d)$$

$$f_0 := 8\epsilon^2 + \frac{4}{1-p}, \quad (5.30e)$$

we define, for all $k \in \{1, \dots, k_\epsilon\}$,

$$B_k := k^p, \quad (5.31a)$$

$$x_k := x_{k-1} + s_{k-1}, \quad (5.31b)$$

$$f_k := f_{k-1} + g_{k-1}s_{k-1}, \quad (5.31c)$$

and for all $k \in \{0, \dots, k_\epsilon\}$,

$$s_k := -B_k^{-1}g_k > 0, \quad (5.32a)$$

$$\nu_k := \frac{1}{\alpha^{-1}\Delta_k^{-1} + |B_k|(1 + \alpha^{-1}\Delta_k^{-1})}. \quad (5.32b)$$

Sequences (5.29), (5.31) and (5.32) may seem obscure without looking at [30, Theorem 2.2.3]. However, they will make more sense in Theorem 2 below. In particular, we aim to have iterates satisfying the assumptions of Proposition 18, along with $\nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = |g_k| > \epsilon$ for

$k \in \{0, \dots, k_\epsilon - 1\}$, and $|g_{k_\epsilon}| = \epsilon$.

First, Lemma 6 establishes bounds on f_k .

Lemma 6. *Using the parameters in (5.28) and the sequences defined in (5.29), (5.31), and (5.32), the following properties hold for the sequence $\{f_k\}$:*

1. for all $k \in \{1, \dots, k_\epsilon\}$,

$$f_k < f_{k-1}, \quad (5.33)$$

2. for all $k \in \{0, \dots, k_\epsilon\}$,

$$0 \leq f_0 - f_k \leq 4\epsilon^2 \left(2 + \frac{k^{(1-p)}}{1-p} \right) \leq 8\epsilon^2 + \frac{4}{1-p}, \quad (5.34)$$

3. for all $k \in \{0, \dots, k_\epsilon\}$,

$$f_k \geq 0. \quad (5.35)$$

Proof. First, we notice that for all $k \in \{0, \dots, k_\epsilon\}$, $g_k < 0$ and $s_k > 0$. By combining these observations and the definition of f_k , we deduce that $f_k < f_{k-1}$ for all $k \in \{0, \dots, k_\epsilon\}$, and in particular

$$f_0 - f_k \geq 0.$$

Inequalities (5.34) hold for $k = 0$ and for $k = 1$ because $f_0 - f_1 = -g_0 s_0 = 4\epsilon^2$. For all $k \in \{2, \dots, k_\epsilon\}$,

$$\begin{aligned} f_0 - f_k &= - \sum_{i=0}^{k-1} g_i s_i \\ &= -g_0 s_0 + \sum_{i=1}^{k-1} g_i^2 i^{-p} \\ &= 4\epsilon^2 + \sum_{i=1}^{k-1} \epsilon^2 (1 + w_i)^2 i^{-p} \\ &= \epsilon^2 \left(4 + \sum_{i=1}^{k-1} (1 + w_i)^2 i^{-p} \right). \end{aligned}$$

Now,

$$\begin{aligned}
\sum_{i=1}^{k-1} (1+w_i)^2 i^{-p} &\leq \sum_{i=1}^{k-1} 4i^{-p} && \text{because } 1+w_i \leq 2 \\
&\leq 4 \left(1 + \sum_{i=2}^{k-1} i^{-p} \right) \\
&\leq 4 \left(1 + \sum_{i=2}^{k-1} \int_{i-1}^i t^{-p} dt \right) && \text{because } i^{-p} = \int_{i-1}^i i^{-p} dt \leq \int_{i-1}^i t^{-p} dt \\
&\leq 4 \left(1 + \int_1^{k-1} t^{-p} dt \right) \\
&\leq 4 \left(1 + \int_1^k t^{-p} dt \right) \\
&= 4 \left(1 + \frac{k^{1-p} - 1}{1-p} \right) \\
&\leq 4 \left(1 + \frac{k^{1-p}}{1-p} \right).
\end{aligned}$$

This results in

$$f_0 - f_k \leq 4\epsilon^2 + 4\epsilon^2 \left(1 + \frac{k^{1-p}}{1-p} \right) = 8\epsilon^2 + 4\frac{\epsilon^2 k^{1-p}}{1-p}. \quad (5.36)$$

Finally, since $k \leq k_\epsilon = \lfloor \epsilon^{-2/(1-p)} \rfloor \leq \epsilon^{-2/(1-p)}$, we have, for all $k \leq k_\epsilon$,

$$\epsilon^2 k^{(1-p)} \leq 1. \quad (5.37)$$

We combine (5.36) and (5.37) to obtain (5.34). The value of f_0 and (5.34) then allows us to establish (5.35). $\square$

Now, Lemma 7 establishes a bound for $|g_{k+1} - g_k|$.

Lemma 7. *Using the parameters in (5.28) and the sequences defined in (5.29), (5.30) and (5.32), we have that, for all $k \in \{0, \dots, k_\epsilon\}$,*

$$|g_{k+1} - g_k| \leq s_k. \quad (5.38)$$

Proof. For $k \in \{0, \dots, k_\epsilon - 1\}$,

$$|g_{k+1} - g_k| = |-\epsilon(1+w_{k+1}) + \epsilon(1+w_k)| = \epsilon/k_\epsilon. \quad (5.39)$$

Since $p < 1$ and $k < k_\epsilon$, we have $k^p/k_\epsilon \leq 1 \leq 1+w_k$. We multiply the latter inequality by

ϵk^{-p} to obtain $\epsilon/k_\epsilon \leq k^{-p}\epsilon(1+w_k)$, which leads to $|g_{k+1} - g_k| \leq s_k$ using (5.39). $\square$

The following result uses Lemma 6 and Lemma 7 to apply Proposition 18.

Proposition 19. *Using the parameters in (5.28) and the sequences defined in (5.29), (5.30) and (5.32), there exists $f : \mathbb{R} \rightarrow \mathbb{R}$ differentiable such that*

$$f(x_k) = f_k, \quad f'(x_k) = g_k. \quad (5.40)$$

Proof. We can see that $s_k > 0$ and, by definition of f_k ,

$$|f_{k+1} - (f_k + g_k s_k)| = 0.$$

Lemma 7 shows that

$$|g_{k+1} - g_k| \leq s_k.$$

Using Lemma 6, we know that for all $k \in \{0, \dots, k_\epsilon\}$, $f_k \geq 0$, and since $\{f_k\}$ is decreasing, we have

$$|f_k| \leq f_0.$$

In addition,

$$|g_k| \leq 2\epsilon \leq 1 \quad \text{and} \quad s_k \leq |g_k| \leq 1.$$

The result follows from Proposition 18. $\square$

For the following lemma, we define the sequence $\{s_{k,1}\}$ such that for all $k \in \{0, \dots, k_\epsilon\}$,

$$s_{k,1} := -\nu_k g_k. \quad (5.41)$$

Lemma 8. *Using the parameters in (5.28) and the sequences defined in (5.29), (5.30) and (5.32), we establish that, for all $k \in \{0, \dots, k_\epsilon\}$,*

$$|s_k| \leq \min(\Delta_k, \beta |s_{k,1}|). \quad (5.42)$$

Proof. On the one hand, we have

$$|s_k| = \epsilon \frac{(1+w_k)}{B_k} \leq 2\epsilon \leq 1 \leq \Delta_k. \quad (5.43)$$

On the other hand, since $B_k^{-1} \leq 1$ and $\Delta_k \geq 1$,

$$2\alpha^{-1} + 1 \geq \alpha^{-1} \Delta_k^{-1} (B_k^{-1} + 1) + 1,$$

so that

$$1 \leq \frac{2\alpha^{-1} + 1}{\alpha^{-1}\Delta_k^{-1}(B_k^{-1} + 1) + 1} \leq \frac{\beta}{\alpha^{-1}\Delta_k^{-1}(B_k^{-1} + 1) + 1}.$$

We multiply the above inequality by B_k^{-1} to obtain

$$B_k^{-1} \leq \frac{\beta B_k^{-1}}{\alpha^{-1}\Delta_k^{-1}(B_k^{-1} + 1) + 1} = \frac{\beta}{\alpha^{-1}\Delta_k^{-1} + B_k(1 + \alpha^{-1}\Delta_k^{-1})} = \beta\nu_k^{-1},$$

and, by multiplying by $|g_k|$, we deduce that

$$|s_k| = B_k^{-1}|g_k| \leq \beta\nu_k^{-1}|g_k| = \beta|s_{k,1}|. \quad (5.44)$$

We combine (5.43) and (5.44) to obtain (5.42). $\square$

The following theorem finally establishes the main result of this section.

Theorem 2 (Slow convergence of Algorithm 5.2.1). *Algorithm 5.2.1 applied to (5.1) with model m_k satisfying Model Assumption 5.2.1, Assumption 2, Assumption 3 and using Hessian approximations $\{B_k\}$ satisfying Assumption 4 may require as many as $O(\epsilon^{-2/(1-p)})$ iterations to produce an iterate x_{k_ϵ} such that*

$$\nu_{k_\epsilon}^{-1/2} \xi_{\text{cp}}(\Delta_{k_\epsilon}; x_{k_\epsilon}, \nu_{k_\epsilon})^{1/2} \leq \epsilon. \quad (5.45)$$

Proof. The proof consists in constructing $f : \mathbb{R} \rightarrow \mathbb{R}$ by interpolation, as in [30, Theorem 2.2.3]. Let $n = 1$, $h = 0$, $\ell = -\infty$, $u = +\infty$. We use the parameters in (5.28) and the sequences defined in (5.29), (5.30) and (5.32). We invoke Proposition 19 to obtain $f : \mathbb{R} \rightarrow \mathbb{R}$ differentiable and bounded such that $f(x_k) = f_k$ and $f'(x_k) = g_k$. Our goal is to show that $\{x_k\}$, $\{s_k\}$, $\{f_k\}$ and $\{g_k\}$ satisfy all our assumptions and are generated by Algorithm 5.2.1 applied to f with $x_0 = 0$ and with the special value of $\{B_k\}$ in (5.30b) and (5.31a).

We proceed by choosing $0 \leq k \leq k_\epsilon$ such that $\Delta_k \geq 1$, which holds at least for $k = 0$, and going through the steps of Algorithm 5.2.1 at iteration k to check that it generates the iterates defined in (5.29), (5.30) and (5.32).

In Line 5, ν_k in (5.32b) is as large as allowed.

In Line 6, Lemma 4 indicates that $s_{k,1}$ in (5.41) is a global minimizer of (5.12b) with $\psi = 0$. As $1 + w_k \leq 2$ and $|B_k| \geq 1$, we observe that

$$|s_{k,1}| = |\nu_k g_k| = \frac{\epsilon(1 + w_k)}{\alpha^{-1}\Delta_k^{-1} + |B_k|(1 + \alpha^{-1}\Delta_k^{-1})} \leq 2\epsilon \leq 1 \leq \Delta_k,$$

which implies that $s_{k,1}$ is a solution of (5.13) because the condition $|s_{k,1}| \leq \Delta_k$ is already satisfied.

In Line 8, let $m_k(\cdot; x_k, B_k)$ be defined as in (5.11). $m_k(\cdot; x_k, B_k)$ satisfies Model Assumption 5.2.1, and using Lemma 4, we have that s_k in (5.32a) with $\psi = 0$ and $B = B_k$ is its global minimizer. Lemma 8 shows that

$$|s_k| \leq \min(\Delta_k, \beta|s_{k,1}|),$$

which also implies that s_k is a solution of (5.11).

In Line 9, we compute

$$\begin{aligned} \rho_k &= \frac{f_k - f_{k+1}}{m(0; x_k, B_k) - m(s_k; x_k, B_k)} \\ &= \frac{f_k - f_{k+1}}{f_k - f_k - g_k s_k - B_k s_k^2 / 2} \\ &= \frac{f_k - f_{k+1}}{g_k^2 B_k^{-1} / 2} \\ &= \frac{-g_k s_k}{g_k^2 B_k^{-1} / 2} \\ &= \frac{B_k^{-1} g_k^2}{g_k^2 B_k^{-1} / 2} \\ &= 2. \end{aligned} \tag{5.46}$$

In Line 10, $\rho_k = 2$ implies that $x_{k+1} = x_k + s_k$, and in Line 11, we can set $\Delta_{k+1} = \min(\gamma_3 \Delta_k, \Delta_{\max}) \geq \Delta_k \geq 1$.

Now, either $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} > \epsilon$, and we perform the next iteration of Algorithm 5.2.1, or $\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} \leq \epsilon$, which stops the algorithm. We have shown that $s_{k,1}$ is a solution of (5.13), thus

$$\xi_{\text{cp}}(\Delta_k; x_k, \nu_k) = f_k - (f_k + g_k s_{k,1}) = -g_k s_{k,1} = \nu_k g_k^2, \tag{5.47}$$

and

$$\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = |g_k|. \tag{5.48}$$

Therefore, for all $k \in \{0, \dots, k_\epsilon - 1\}$,

$$\nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} > \epsilon,$$

and

$$\nu_{k_\epsilon}^{-1/2} \xi_{\text{cp}}(\Delta_{k_\epsilon}; x_{k_\epsilon}, \nu_{k_\epsilon})^{1/2} = \epsilon,$$

so that Algorithm 5.2.1 performs exactly k_ϵ iterations to generate x_{k_ϵ} satisfying (5.45).

To finish the proof, we must verify that Assumption 2, Assumption 3 and Assumption 4 hold. Assumption 2 is satisfied thanks to Proposition 16. Assumption 3 is satisfied with $\kappa_{\text{ubd}} = \frac{1}{2}$ because

$$|f_{k+1} - m(s_k; x_k, B_k)| = |f_{k+1} - f_k - g_k s_k - \frac{1}{2} B_k s_k^2| = \frac{1}{2} B_k s_k^2 \leq \frac{1}{2} (1 + B_k) s_k^2.$$

Finally, our choice of B_k allows Assumption 4 to be satisfied because all iterations are successful and $\sigma_k = k$. $\square$

5.5 Numerical verification of the bound

We construct $f : \mathbb{R} \rightarrow \mathbb{R}$ satisfying the properties of the function in the proof of Theorem 2. The construction follows the formula used in the proof of [30, Theorem A.9.2], and we use similar notation.

We use again the parameters (5.28), and the sequences (5.29)–(5.32). Define the cubic Hermite interpolant

$$\pi_k(\tau) := c_{k,0} + c_{k,1}\tau + c_{k,2}\tau^2 + c_{k,3}\tau^3, \quad (5.49)$$

where, for all $k \in \{0, \dots, k_\epsilon\}$, $c_{k,0} = f_k$, $c_{k,1} = g_k$, and $c_{k,2}, c_{k,3}$ solve

$$\begin{bmatrix} s_k^2 & s_k^3 \\ 2s_k & 3s_k^2 \end{bmatrix} \begin{bmatrix} c_{k,2} \\ c_{k,3} \end{bmatrix} = \begin{bmatrix} f_{k+1} - (f_k + g_k s_k) \\ g_{k+1} - g_k \end{bmatrix} = \begin{bmatrix} 0 \\ g_{k+1} - g_k \end{bmatrix}. \quad (5.50)$$

We use the additional conditions $f_{-1} = f_0$, $g_{-1} = 0$, $f_{k_\epsilon+1} = f_{k_\epsilon}$, $g_{k_\epsilon+1} = g_{k_\epsilon}$, and $x_{-1} = -s_{-1}$, where $s_{-1} = 1$, which allows (5.27) to hold with $\kappa_f = 1$, because $|f_0 - (f_{-1} + g_{-1}s_{-1})| = 0$, and $|g_0 - g_{-1}| = |g_0| = \epsilon(1 + w_0) = 2\epsilon \leq 1 = s_{-1}$ since $\epsilon \leq 1/2$. Finally,

$$f(x) := \begin{cases} f_0 & \text{if } x \leq x_{-1} \\ \pi_k(x - x_k) & \text{if } x \in (x_k; x_{k+1}] \text{ for } k \in \{-1, \dots, k_\epsilon\} \\ f_{k_\epsilon} & \text{if } x > x_{k_\epsilon} + s_{k_\epsilon}. \end{cases} \quad (5.51)$$

By construction, f is a piecewise polynomial of degree 3. We have $\pi_k(0) = f_k$, $\pi'_k(0) = g_k$, $\pi_k(s_k) = f_{k+1}$ thanks to the definition of f in (5.31c) and the first line of (5.50), and $\pi'_k(s_k) = g_{k+1}$ with the second line of (5.50). Thus, $f : \mathbb{R} \rightarrow \mathbb{R}$ is continuously differentiable over $(x_{-1}, x_{k_\epsilon+1})$.

We minimize f using Algorithm 5.2.1 as implemented in [10], without nonsmooth regularizer, and with starting point $x_0 = 0$. Inside TR, we set $B_k = k^p$ so that $\{B_k\}$ grows unbounded and Assumption 4 holds, because $\rho_k = 2$ in (5.46) so that all iterations are very successful. In Line 8, we use the analytical solution $s_k = -B_k^{-1}\nabla f(x_k)$ of (5.17) given by Lemma 4 in order to avoid rounding errors occurring in a subproblem solver for (5.15). This expression of s_k satisfies the trust-region constraint by construction thanks to Lemma 8. The modified TR implementation is available from <https://github.com/geoffroyleconte/RegularizedOptimization.jl/tree/unbounded>.

We set $p = 1/10$, $\alpha = \beta = 10^{+16}$, $\gamma_3 = 3$, $\Delta_{\max} = 10^3$ and $\epsilon = 1/10$, so that $k_\epsilon = 166$. We observe that TR converges in precisely 166 iterations. With $\epsilon = 1/20$, we obtain the convergence of TR in precisely $k_\epsilon = 778$ iterations.

In order to make the oscillations of f' clearly visible, Figure 5.2 shows plots of f and f' over $[0, x_{k_\epsilon+1}]$ with $\epsilon = 1/3$. Table 5.1 shows the theoretical values of $\nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = |g_k|$ according to (5.48). TR converges in 11 iterations and produces the logs in Figure 5.1 that align with these theoretical values. Note that $\rho_k = 2$, as predicted by (5.46), and therefore, that each iteration is successful.

Table 5.1 Rounded theoretical values of $\nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$ for $\epsilon = 1/3$.

k	0	1	2	3	4	5	6	7	8	9	10	11
$\nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$	0.67	0.64	0.61	0.58	0.55	0.52	0.48	0.45	0.42	0.39	0.36	0.33

outer	inner	f(x)	h(x)	$\sqrt{\xi_{\text{cp}}}/\sqrt{\nu}$	$\sqrt{\xi}$	ρ	Δ	$\ x\ $	$\ s\ $	$\ B_k\ $
1	1	5.3e+00	0.0e+00	6.7e-01	4.7e-01	2.0e+00	1.0e+00	0.0e+00	6.7e-01	1.0e+00
2	1	4.9e+00	0.0e+00	6.4e-01	4.5e-01	2.0e+00	3.0e+00	6.7e-01	6.4e-01	1.0e+00
3	1	4.5e+00	0.0e+00	6.1e-01	4.1e-01	2.0e+00	9.0e+00	1.3e+00	5.7e-01	1.1e+00
4	1	4.1e+00	0.0e+00	5.8e-01	3.9e-01	2.0e+00	2.7e+01	1.9e+00	5.2e-01	1.1e+00
5	1	3.8e+00	0.0e+00	5.5e-01	3.6e-01	2.0e+00	8.1e+01	2.4e+00	4.7e-01	1.1e+00
6	1	3.6e+00	0.0e+00	5.2e-01	3.4e-01	2.0e+00	2.4e+02	2.9e+00	4.4e-01	1.2e+00
7	1	3.4e+00	0.0e+00	4.8e-01	3.1e-01	2.0e+00	7.3e+02	3.3e+00	4.1e-01	1.2e+00
8	1	3.2e+00	0.0e+00	4.5e-01	2.9e-01	2.0e+00	1.0e+03	3.7e+00	3.7e-01	1.2e+00
9	1	3.0e+00	0.0e+00	4.2e-01	2.7e-01	2.0e+00	1.0e+03	4.1e+00	3.4e-01	1.2e+00
10	1	2.8e+00	0.0e+00	3.9e-01	2.5e-01	2.0e+00	1.0e+03	4.4e+00	3.2e-01	1.2e+00
11	1	2.7e+00	0.0e+00	3.6e-01	2.3e-01	2.0e+00	1.0e+03	4.7e+00	2.9e-01	1.3e+00
12	1	2.6e+00	0.0e+00	3.3e-01			1.0e+03	5.0e+00	2.6e-01	1.3e+00

TR: terminating with $\sqrt{\xi_{\text{cp}}}/\sqrt{\nu} = 0.3333333333333333$
"Execution stats: first-order stationary"

Figure 5.1 TR logs with $\epsilon = 1/3$. *outer* denotes the iteration number, *inner* is the number of iterations performed by the subsolver to solve (5.15) with the model in (5.11), $\sqrt{\xi_{\text{cp}}}/\sqrt{\nu}$ is $\nu_k^{-1/2}\xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2}$, $\sqrt{\xi}$ is the numerator of (5.16), $\|s\|$ is $\|s_k\|$, and the remaining columns refer unambiguously to data used in Algorithm 5.2.1.

The code to run this experiment is available at <https://github.com/geoffroyleconte/docGL/blob/master/regularized-opt/test-unbounded-hess.jl>. By making similar changes to the algorithm TRDH [54], which can be found at the same URL, we obtain the same number of iterations.

5.6 Discussion

We have shown that it is possible to establish convergence and sharp worst-case evaluation complexity of Algorithm 5.2.1 in the presence of unbounded Hessian approximations B_k , provided they do not grow too fast—c.f., Assumption 4. We established that the complexity bound can be attained, and we gave an example of function for which it was attained, both theoretically and numerically.

When $p \geq 1$ in Assumption 4 or the growth of $\|B_k\|$ is not governed by the number of successful iterations, it may still be possible to establish convergence in the sense that $\liminf \nu_k^{-1/2} \xi_{\text{cp}}(\Delta_k; x_k, \nu_k)^{1/2} = 0$ as in [34, §8.4.1.2], where the main assumption is that

$$\sum_{k=0}^{\infty} \frac{1}{1 + \max_{0 \leq j \leq k} \|B_j\|} = \infty.$$

However, it is unclear at the time of this writing whether a sharp worst-case evaluation complexity bound holds for such more general cases.

Competing interests

We certify that the research submitted here is original, is our own, and is not being evaluated elsewhere for publication. This work was supported by an NSERC Discovery grant.

Data availability

The code used to produce the numerical results is available from <https://github.com/geoffroyleconte/unbounded-hessian-code>. The solvers are available from <https://github.com/geoffroyleconte/RegularizedOptimization.jl/tree/unbounded>.

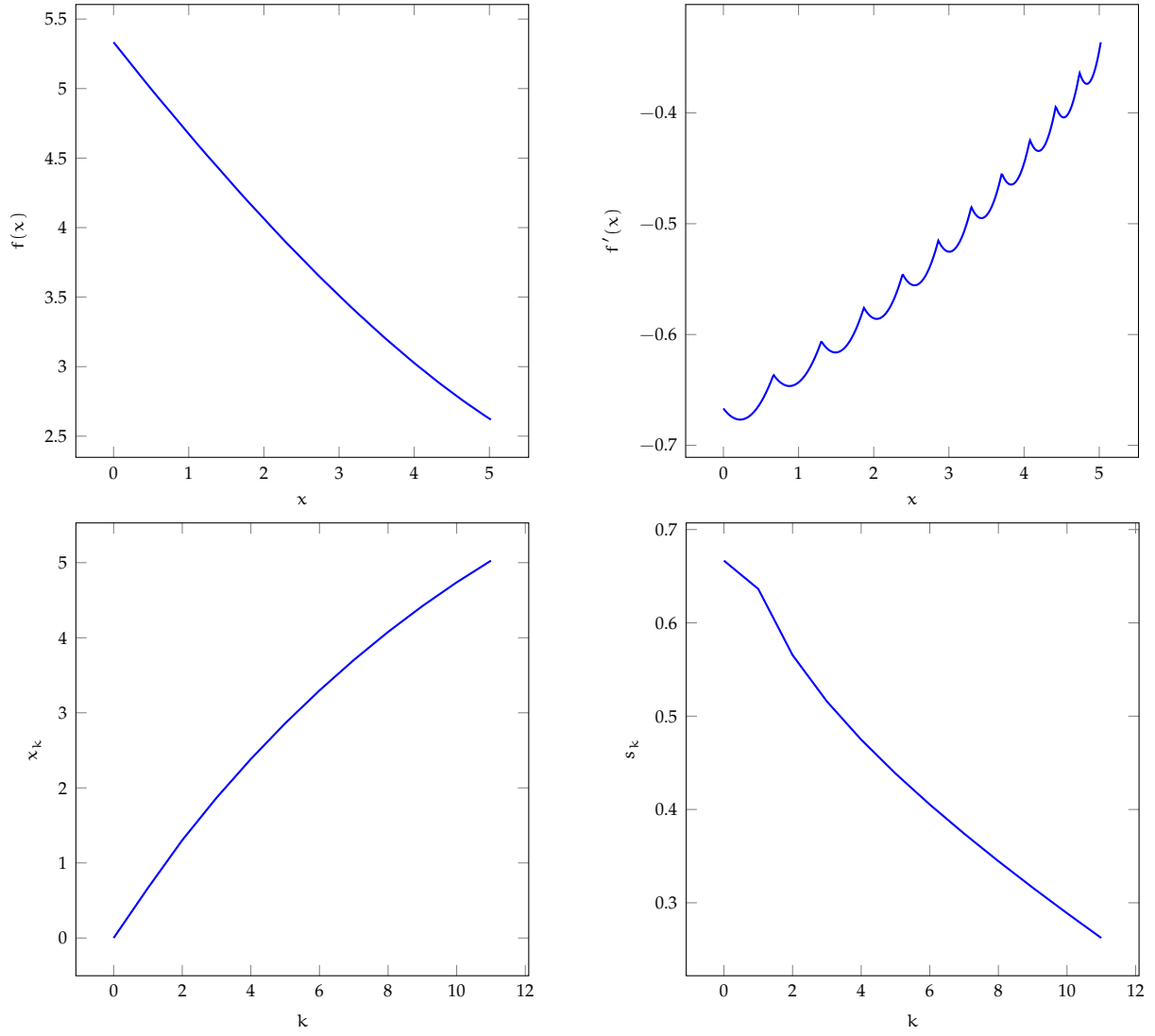


Figure 5.2 Illustration of example (5.51) with $\epsilon = 1/3$. Top row: values of f (left) and of f' (right) for $x \in [0, x_{k_\epsilon+1}]$. Bottom row: iterates x_k (left) and steps s_k (right) for $k \in [0, k_\epsilon + 1]$.

CHAPITRE 6 ARTICLE 3: AN INTERIOR-POINT TRUST-REGION METHOD FOR NONSMOOTH REGULARIZED BOUND-CONSTRAINED OPTIMIZATION

Nous nous intéressons à la résolution de problèmes d’optimisation non lisse avec contraintes de bornes tels que (2.72). Nous avons déjà évoqué que TR, TRDH et R2 peuvent être utilisés pour résoudre (2.72) (et nous le rappelons de manière plus approfondie dans ce chapitre). De telles méthodes utilisées pour résoudre (2.72) sont parfois appelées méthodes projetées, voir par exemple [34, Chapitre 12] pour les méthodes de région de confiance projetées dans le cas lisse. Ce chapitre [56] a pour but de développer un algorithme de points intérieurs basé sur une méthode de région de confiance proximale, de prouver sa convergence, et de le comparer aux méthodes projetées. Nous nous basons en partie sur l’algorithme présenté dans [34, Chapitre 13.6] dans le cas lisse.

Notre algorithme est appelé RIPM pour *Regularized Interior Proximal Method* en anglais, et il résout une suite de sous-problèmes barrière non contraints en utilisant un algorithme de région de confiance inspiré de TR. Nous implémentons également un algorithme appelé RIPMDH pour *Regularized Interior Proximal Method with Diagonal Hessian approximations* en anglais, qui résout les sous-problèmes barrière avec une méthode similaire à TRDH présentée dans le chapitre 4. Sous une hypothèse supplémentaire portant sur la convexité du terme de régularisation, nous implémentons une variante de RIPM et RIPMDH basée sur un critère d’arrêt différent de celui de TR, permettant d’obtenir de meilleures performances.

Afin de ne pas rendre les démonstrations trop compliquées, nous considérons le cas particulier

$$\min_x f(x) + h(x) \quad \text{s.c.} \quad x \geq 0,$$

mais nos algorithmes sont en fait capables de résoudre (2.72) pour des raisons qui seront évoquées plus tard dans ce chapitre.

Title: An interior-point trust-region method for nonsmooth regularized bound-constrained optimization

Authors: Geoffroy Leconte and Dominique Orban

Submitted to: SIAM journal on Optimization

Submission date: 28/02/2024

Abstract: We develop an interior-point method for nonsmooth regularized bound-constrained optimization problems. Our method consists of iteratively solving a sequence of unconstrained nonsmooth barrier subproblems. We use a variant of the proximal quasi-Newton trust-region algorithm TR of Aravkin *et al.* [4] to solve the barrier subproblems, with additional assumptions inspired from well-known smooth interior-point trust-region methods. We show global convergence of our algorithm with respect to the criticality measure of Aravkin *et al.* [4]. Under an additional assumption linked to the convexity of the nonsmooth term in the objective, we present an alternative interior-point algorithm with a slightly modified criticality measure, which performs better in practice. Numerical experiments show that our algorithm performs better than the trust-region method TR, the trust-region method with diagonal hessian approximations TRDH of Leconte and Orban [54], and the quadratic regularization method R2 of Aravkin *et al.* [4] for two out of four tested bound-constrained problems. On those two problems, our algorithm obtains smaller objective values than the other solvers using fewer objective and gradient evaluations. On the two other problems, it performs similarly to TR, R2 and TRDH.

Keywords: Regularized optimization, nonsmooth optimization, nonconvex optimization, bound constraints, proximal gradient method, barrier method.

AMS subject classifications: 49J52, 65K10, 90C53, 90C56,

6.1 Introduction

We consider the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ f(x) + h(x) \quad \text{subject to } x \geq 0, \quad (6.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has Lipschitz-continuous gradient with constant $L_f \geq 0$ and $h : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper and lower semi-continuous. Both f and h may be nonconvex. h is often considered as a regularization function used to favor solutions with desirable properties, such as sparsity.

Problems such as (6.1) are classically solved with a variant of the proximal gradient method [61]. The proximal quasi-Newton trust-region algorithm of Aravkin *et al.* [4], referred to as TR, can be extended to box constraints, provided that the proximal operator of $h + \chi(\cdot \mid [\ell, u])$, where $\ell < u$ componentwise and χ is the indicator of the box $[\ell, u]$, can be computed efficiently. Leconte and Orban [54] present a variant of TR named TRDH that also supports

box constraints, and uses diagonal quasi-Newton approximations. For specific separable regularizers h , they provide a closed-form solution of the trust-region subproblems with box constraints, giving rise to what they defined as an indefinite proximal operator; a scaled generalization of a proximal operator.

At each iteration k , we solve the barrier subproblem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ f(x) + \phi_k(x) + h(x), \quad (6.2)$$

where ϕ_k is the logarithmic barrier function

$$\phi_k(x) = \sum_{i=1}^n \phi_{k,i}(x), \quad \phi_{k,i}(x) := -\mu_k \log(x_i), \ i = 1, \dots, n, \quad (6.3)$$

and $\{\mu_k\} \searrow 0$. Each subproblem (6.2) is an unconstrained problem if we consider that $-\log(x) = +\infty$ when $x \leq 0$. In Section 6.5.2, we explain that under reasonable assumptions, we can solve (6.2) with a modified version of Aravkin *et al.*'s TR algorithm, and we expect that the solutions of (6.3) converge to a solution of (6.1) as $\mu_k \rightarrow 0$.

Our approach is sometimes referred to as a trust-region interior-point method, or trust-region method for barrier functions. We refer the reader to [34, Chapter 13] for more information on the case where $h = 0$. Our algorithm, named RIPM (*Regularized Interior Proximal Method*), can be seen as a generalization of those methods to solve (6.1).

An inconvenience of solving (6.2), induced by the logarithmic barrier function (6.3), is that the smooth part of the subproblem $f + \phi_k$ does not have a Lipschitz gradient, thus compromising the convergence properties of TR established by Aravkin *et al.* [4]. Nevertheless, in our analysis, we establish the convergence of the barrier subproblems using the update rules of Conn *et al.* [34, Chapter 13.6.3] for our trust-region model. We also show global convergence of RIPM towards a first-order stationary point of (6.1) if the trust-region radii and the step lengths used in proximal operator evaluations are bounded away from zero, and the iterates generated by the algorithm remain bounded. In Section 6.5.4, under a convexity assumption on the nonsmooth term, we provide an alternative implementation of the outer iterations of RIPM where we change the stopping criteria to improve numerical performance.

In addition, we implement a variant of RIPM named RIPMDH (*Regularized Interior Proximal Method with Diagonal Hessian approximations*) that uses TRDH to solve the barrier subproblems. We compare the performance of RIPM and RIPMDH with TR, TRDH and R2, all available from RegularizedOptimization.jl [10], on four bound-constrained problems. The first two problems are a regularized box-constrained quadratic problem, and a sparse

nonnegative matrix factorization problem. These two problems require many TR and R2 iterations to converge. RIPM and RIPMDH obtain smaller objective values than the other solvers using fewer objective and gradient evaluations, which suggests that they may be best suited to solve difficult bound-constrained nonsmooth problems. The third problem is an inverse problem for finding the parameters of a differential equation. RIPM and RIPMDH perform more objective and gradient evaluations than TR, but RIPMDH performs the least amount of proximal operator evaluations. The last problem is a regularized box-constrained basis-pursuit denoise problem. RIPMDH exhibits similar performance to those of TR, TRDH and R2 using a modification of some of its parameters.

Related research

Attouch and Wets [6] use $f = 0$ and the nonsmooth barrier $-\mu_k \sum_{i=1}^n \log(\min(\frac{1}{2}, x_i)) > 0$. They use the theory of epi-convergence to explain some convergence properties of barrier methods, and in particular, that the objectives of the barrier subproblems epi-converge to the objective of their initial constrained problem.

Chouzenoux *et al.* [32] use convex f and h with general inequality constraints $c_i(x) \leq 0$ for $i \in \{1, \dots, p\}$, $p > 0$ to solve large-scale image processing problems. Their algorithm uses proximal gradient steps to solve the barrier subproblem.

Bertocchi *et al.* [15] solve inverse problems $y = \mathcal{D}(H\bar{x})$, where y is some observed data, $\bar{x}$ the signal to determine, H is a linear observation operator, and $\mathcal{D}$ is an operator applying noise perturbation. They solve the problem $\min_{x \in \mathbb{R}^n} f(Hx, y) + h(x)$ with constraints $c_i(x) \geq 0, i \in \{1, \dots, p\}$, where f is a convex function involving the observations and the signal x , $f(\cdot; y)$ and h are twice differentiable, h and $-c_i$ are convex (among other properties). They compute the proximal operator of the barrier term, and use an interior-point algorithm. They apply their algorithm to develop a neural network architecture for image restoration.

De Marchi and Themelis [37] use the proximal gradient method to solve nonsmooth regularized optimization problems such as (6.1) where f has a locally Lipschitz-continuous gradient, h is continuous relative to its domain and prox-bounded. In addition, the constraint $x \geq 0$ is replaced by a more general constraint $c(x) \leq 0$, where c has locally Lipschitz-continuous Jacobian.

Shen *et al.* [75] present an active set proximal algorithm to solve (6.1) with $h(x) = \lambda \|x\|_1$ for some $\lambda > 0$, and where $\ell \leq x \leq u$ with $\ell < 0 < u$ instead of $x \geq 0$. They use a hybrid search direction based upon a proximal gradient step for the active variables (i.e., the variables that are at one the bounds of the constraints), and a Newton step for the other variables.

Notation

For $v \in \mathbb{R}^n$, $\|v\|$ denotes the Euclidean norm of v . $\mathbb{R}_+$ and $\mathbb{R}_{++}$, denote, respectively, the sets of positive and strictly positive real numbers, whereas $\mathbb{R}_+^n$ and $\mathbb{R}_{++}^n$ denote the sets of vectors having all their components in $\mathbb{R}_+$ and $\mathbb{R}_{++}$, respectively.

$\bar{\mathbb{R}}$ denotes $\mathbb{R} \cup \{\pm\infty\}$. The unit closed ball defined with the ℓ_∞ -norm and centered at the origin is $\mathbb{B}$, and the ball centered at the origin of radius $\Delta > 0$ is $\Delta\mathbb{B}$. If $C \subseteq \mathbb{R}^n$, the indicator of C is $\chi(\cdot \mid C) : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ defined by $\chi(x \mid C) = 0$ if $x \in C$, and $\chi(x \mid C) = +\infty$ if $x \notin C$. For $y \in \mathbb{R}^n$, the set $y + C$ is composed of all the vectors $s \in \mathbb{R}^n$ such that $s = y + x$ with $x \in C$.

Following the notation of Rockafellar and Wets [73], the set of all subsequences of $\mathbb{N}$ is denoted by $\mathcal{N}_\infty^\#$, and the set composed of the subsequences of $\mathbb{N}$ containing all k beyond some k_0 is denoted by $\mathcal{N}_\infty$. For $N \in \mathcal{N}_\infty^\#$, $\{x_k\} \xrightarrow[N]{} \bar{x}$ indicates that the subsequence $\{x_k\}_{k \in N}$ (which we may also write $\{x_k\}_N$ for conciseness) converges to $\bar{x}$.

X , Z and S (possibly with subscripts k , k,j or $k,j,1$) denote the square diagonal matrices having x , z and s as diagonal elements, respectively.

6.2 Background

The following are standard variational analysis concepts—see, e.g., [73]. Let $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $\bar{x} \in \mathbb{R}^n$ where ϕ is finite. The Fréchet subdifferential of ϕ at $\bar{x}$ is the closed convex set $\hat{\partial}\phi(\bar{x})$ of elements $v \in \mathbb{R}^n$ such that

$$\liminf_{\substack{x \rightarrow \bar{x} \\ x \neq \bar{x}}} \frac{\phi(x) - \phi(\bar{x}) - v^T(x - \bar{x})}{\|x - \bar{x}\|} \geq 0.$$

The limiting subdifferential of ϕ at $\bar{x}$ is the closed, but not necessarily convex, set $\partial\phi(\bar{x})$ of elements $v \in \mathbb{R}^n$ for which there exist $\{x_k\} \rightarrow \bar{x}$ and $\{v_k\} \rightarrow v$ such that $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$ and $v_k \in \hat{\partial}\phi(x_k)$ for all k . The inclusion $\hat{\partial}\phi(\bar{x}) \subset \partial\phi(\bar{x})$ always holds. Finally, the horizon subdifferential of ϕ at $\bar{x}$ is the closed, but not necessarily convex, cone $\partial^\infty\phi(\bar{x})$ of elements $v \in \mathbb{R}^n$ for which there exist $\{x_k\} \rightarrow \bar{x}$, $\{v_k\}$ and $\{\lambda_k\} \searrow 0$ such that $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$, $v_k \in \hat{\partial}\phi(x_k)$ for all k , and $\{\lambda_k v_k\} \rightarrow v$.

If $\inf \phi > -\infty$, $\operatorname{argmin} \phi$ is the set of $x \in \mathbb{R}^n$ such that $\phi(x) = \inf \phi$. For $\epsilon > 0$, ϵ - $\operatorname{argmin} \phi$ is the set of $x \in \mathbb{R}^n$ such that $\phi(x) \leq \inf \phi + \epsilon$.

If $C \subseteq \mathbb{R}^n$ and $\bar{x} \in C$, the closed convex cone $\widehat{N}_C(\bar{x}) := \widehat{\partial}\chi(\bar{x} \mid C)$ is the regular normal cone to C at $\bar{x}$. The closed cone $N_C(\bar{x}) := \partial\chi(\bar{x} \mid C) = \partial^\infty\chi(\bar{x} \mid C)$ is the normal cone to C at $\bar{x}$. $\widehat{N}_C(\bar{x}) \subseteq N_C(\bar{x})$ always holds, and is an equality if C is convex.

If C is convex, $N_C(\bar{x})$ is the closed convex cone of elements $v \in \mathbb{R}^n$ such that $v^T(x - \bar{x}) \leq 0$ for all $x \in C$ [73, Theorem 6.9].

For a set-valued mapping $S : X \rightarrow U$ where, for any $x \in X$, $S(x) \subset U$, the graph of S is the set $\text{gph } S := \{(x, u) \mid u \in S(x)\}$.

For $\bar{x} \in \mathbb{R}^n$, the limit superior of S at $\bar{x}$ is

$$\limsup_{x \rightarrow \bar{x}} S(x) := \{u \mid \exists \{x_k\} \rightarrow \bar{x}, \exists \{u_k\} \rightarrow u \text{ with } u_k \in S(x_k)\},$$

and the limit inferior of S at $\bar{x}$ is

$$\liminf_{x \rightarrow \bar{x}} S(x) := \{u \mid \forall \{x_k\} \rightarrow \bar{x}, \exists N \in \mathcal{N}_\infty, \{u_k\} \xrightarrow{N} u \text{ with } u_k \in S(x_k)\}.$$

$S(\bar{x}) \subseteq \limsup_{x \rightarrow \bar{x}} S(x)$ and $S(\bar{x}) \supseteq \liminf_{x \rightarrow \bar{x}} S(x)$ always hold.

The set-valued mapping S is outer semicontinuous (osc) at $\bar{x}$ if $\limsup_{x \rightarrow \bar{x}} S(x) \subseteq S(\bar{x})$, or, equivalently, $\limsup_{x \rightarrow \bar{x}} S(x) = S(\bar{x})$. It is inner semicontinuous (isc) at $\bar{x}$ if $\liminf_{x \rightarrow \bar{x}} S(x) \supseteq S(\bar{x})$, or equivalently $\liminf_{x \rightarrow \bar{x}} S(x) = S(\bar{x})$ when S is closed-valued. If both conditions hold, S is continuous at $\bar{x}$, i.e., $S(x) \rightarrow S(\bar{x})$ as $x \rightarrow \bar{x}$.

Proposition 20 (73, Proposition 8.7). *For $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $\bar{x}$ where ϕ is finite, $\partial\phi$ is osc at $\bar{x}$ with respect to $\phi(x) \rightarrow \phi(\bar{x})$ when $x \rightarrow \bar{x}$, i.e. for any $\{x_k\} \rightarrow \bar{x}$ with $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$, there exists $v_k \in \partial\phi(x_k)$ for all k such that $\{v_k\} \rightarrow \bar{v} \in \partial\phi(\bar{x})$.*

The graphical outer limit of a sequence of set-valued mappings S_k is defined by

$$(\text{g-lim}_k \sup S_k)(x) := \{u \mid \exists N \in \mathcal{N}_\infty^\#, \{x_k\} \xrightarrow{N} x, \{u_k\} \xrightarrow{N} u, u_k \in S_k(x_k)\}.$$

The graphical inner limit of a sequence of set-valued mappings S_k is defined by

$$(\text{g-lim}_k \inf S_k)(x) := \{u \mid \exists N \in \mathcal{N}_\infty, \{x_k\} \xrightarrow{N} x, \{u_k\} \xrightarrow{N} u, u_k \in S_k(x_k)\}.$$

If both limits agree, the graphical limit $S = \text{g-lim } S_k$ exists, so that we can also write $S_k \xrightarrow{\text{g}} S$, and we have $S_k \xrightarrow{\text{g}} S \iff \text{gph } S_k \rightarrow \text{gph } S$.

The epigraph of ϕ is the set $\text{epi } \phi = \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid \alpha \geq \phi(x)\}$.

We denote $\text{cl}(\phi)$ the (lower) closure of ϕ , i.e., the largest function less than ϕ that is lower semi-continuous. Its epigraph is the closure of $\text{epi } \phi$.

If $\phi_k : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ for all $k \geq 0$, the lower and upper pointwise limits of $\{\phi_k\}$ are the functions

$\text{p-lim inf}_k \phi_k$ and $\text{p-lim sup}_k \phi_k : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ defined for all $x \in \mathbb{R}^n$ by

$$\begin{aligned} (\text{p-lim inf}_k \phi_k)(x) &:= \liminf_k \phi_k(x), \\ (\text{p-lim sup}_k \phi_k)(x) &:= \limsup_k \phi_k(x). \end{aligned}$$

When $\text{p-lim inf}_k \phi_k$ and $\text{p-lim sup}_k \phi_k$ coincide, their common value is the pointwise limit of $\{\phi_k\}$ denoted $\text{p-lim}_k \phi_k$. If $\text{p-lim}_k \phi_k = \phi$, we write $\{\phi_k\} \xrightarrow{\text{p}} \phi$.

For a sequence $\{E_k\} \subseteq \mathbb{R}^n$, we define

$$\begin{aligned} \limsup_k E_k &:= \{z \in \mathbb{R}^n \mid \exists N \in \mathcal{N}_\infty^\#, \exists \{z_j\}_N \rightarrow z, z_j \in E_j \text{ for all } j \in N\} \\ \liminf_k E_k &:= \{z \in \mathbb{R}^n \mid \exists N \in \mathcal{N}_\infty, \exists \{z_j\}_N \rightarrow z, z_j \in E_j \text{ for all } j \in N\}. \end{aligned}$$

Consider in particular $E_k := \text{epi } \phi_k$. It is not difficult to see that $\limsup_k E_k$ and $\liminf_k E_k$ are also epigraphs in the sense that if (x, t) is in either set, then so is (x, s) for any $s \geq t$. The lower epi-limit of $\{\phi_k\}$ is the function $\text{e-lim inf}_k \phi_k$ whose epigraph is $\limsup_k E_k$, and the upper epi-limit of $\{\phi_k\}$ is the function $\text{e-lim sup}_k \phi_k$ whose epigraph is $\liminf_k E_k$. It is always true that $\text{e-lim inf}_k \phi_k \leq \text{e-lim sup}_k \phi_k$. When the two coincide, their common value is called the epi-limit of $\{\phi_k\}$ denoted $\text{e-lim}_k \phi_k$. If $\text{e-lim}_k \phi_k = \phi$, we also write $\{\phi_k\} \xrightarrow{\text{e}} \phi$.

The following result summarizes important properties of epi-limits used in the sequel.

Proposition 21 (73, Proposition 7.4). *Let $\phi_k : \mathbb{R}^n \rightarrow \mathbb{R}$ for $k \geq 0$.*

1. *$\text{e-lim inf}_k \phi_k$ and $\text{e-lim sup}_k \phi_k$ are lsc, and so is $\text{e-lim}_k \phi_k$ when it exists;*
2. *if $\phi_k \geq \phi_{k+1}$ for all k , $\text{e-lim}_k \phi_k$ exists and equals $\text{cl}(\inf_k \phi_k)$;*
3. *if $\phi_k \leq \phi_{k+1}$ for all k , $\text{e-lim}_k \phi_k$ exists and equals $\text{cl}(\sup_k \phi_k)$.*

In addition, if $\phi, \underline{\phi}_k, \bar{\phi}_k : \mathbb{R}^n \rightarrow \mathbb{R}$ with $\underline{\phi}_k \leq \phi_k \leq \bar{\phi}_k$ for all k , and if $\{\underline{\phi}_k\} \xrightarrow{\text{e}} \phi$ and $\{\bar{\phi}_k\} \xrightarrow{\text{e}} \phi$, then $\{\phi_k\} \xrightarrow{\text{e}} \phi$.

The model that we will use in our algorithm uses an approximation $\psi(\cdot; x)$ of h at x so that $\psi(s, x) \approx h(x + s)$. For $\psi : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \overline{\mathbb{R}}$, the function-valued mapping $x \mapsto \psi(\cdot; x)$ is *epi-continuous* at $\bar{x}$ if $\psi(\cdot, x) \xrightarrow{\text{e}} \psi(\cdot, \bar{x})$ as $x \rightarrow \bar{x}$.

ϕ is level-bounded if, for every $\alpha \in \mathbb{R}$, the lower level set $\text{lev}_{\leq \alpha} \phi := \{x \in \mathbb{R}^n \mid \phi(x) \leq \alpha\}$ is bounded (possibly empty). The sequence of functions $\{\phi_k\}$ is eventually level-bounded if, for each $\alpha \in \mathbb{R}$, the sequence of sets $\{\text{lev}_{\leq \alpha} \phi_k\}$ is eventually bounded, i.e., there is an index set $N \in \mathcal{N}_\infty$ such that $\{\text{lev}_{\leq \alpha} \phi_k\}_{k \in N}$ is bounded.

The following theorem establishes properties about the minimization of sequences of epi-convergent functions.

Theorem 3 (73, Theorem 7.33). *Suppose the sequence $\{\phi_k\}$ is eventually level-bounded, and $\phi_k \xrightarrow{e} \phi$ with ϕ_k and ϕ lsc and proper. Then,*

$$\inf \phi_k \rightarrow \inf \phi \quad (6.4)$$

with $-\infty < \inf \phi < +\infty$, while there exists $N \in \mathcal{N}_\infty$ such that $\operatorname{argmin} \phi_k$ is a bounded sequence of nonempty sets with

$$\limsup_k (\operatorname{argmin} \phi_k) \subset \operatorname{argmin} \phi. \quad (6.5)$$

Indeed, for any $\{\epsilon_k\} \searrow 0$ and $x_k \in \epsilon_k$ - $\operatorname{argmin} \phi_k$, $\{x_k\}$ is bounded and all its cluster points belong to $\operatorname{argmin} \phi$. If $\operatorname{argmin} \phi$ consists of a unique point $\bar{x}$, one must actually have $\{x_k\} \rightarrow \bar{x}$.

The proximal operator associated with the proper lsc function h and parameter $\nu > 0$ is

$$\operatorname{prox}_{\nu h}(x) := \arg \min_w \frac{1}{2} \nu^{-1} \|w - x\|^2 + h(w). \quad (6.6)$$

6.3 Stationarity

First-order stationarity conditions for (6.1) may be stated as [73, Theorem 10.1]

$$0 \in \nabla f(x) + \partial(h + \chi(\cdot \mid \mathbb{R}_+^n))(x). \quad (6.7)$$

We say that the constraint qualification (CQ) holds at x if $\partial^\infty(f + h)(x)$ contains no $v \neq 0$ such that $-v \in N_{\mathbb{R}_+^n}(x)$.

Under the constraint qualification, (6.7) can also be written [73, Theorem 8.15]

$$0 \in \nabla f(x) + \partial h(x) + N_{\mathbb{R}_+^n}(x). \quad (6.8)$$

Our assumptions that f is continuously differentiable and that h is proper lsc allows us to write [73, Exercise 8.8 and Theorem 8.9]

$$\partial^\infty(f + h)(x) = \partial^\infty h(x) = \{v \in \mathbb{R}^n \mid (v, 0) \in N_{\operatorname{epi} h}(x, h(x))\}.$$

Due to the simple form of $N_{\mathbb{R}_+^n}$, the constraint qualification states that the only $v \in \partial^\infty h(x)$ such that $v \geq 0$ and satisfying $v_i = 0$ if $x_i > 0$ is $v = 0$.

A simple example where the constraint qualification is not satisfied is found by setting $n = 1$, $f = 0$, and $h(x) = |x|_0$, i.e., $h(x) = 1$ if $x \neq 0$ and $h(0) = 0$, in (6.1). The unique solution is $\bar{x} = 0$. Then, $\widehat{N}_{\text{epi}h}(0, 0) = N_{\text{epi}h}(0, 0) = \{(v, t) \mid t \leq 0\}$ so that $\partial^\infty h(0) = \mathbb{R}$. The qualification condition requires that the only $v \in \mathbb{R}$ such that $v \geq 0$ be $v = 0$, which is clearly not the case. Of course, the bound constraint in the example above is redundant and the constrained and unconstrained solutions coincide.

Using [73, Example 6.10], $N_{\mathbb{R}_+^n}(x) = N_{\mathbb{R}_+}(x_1) \times \cdots \times N_{\mathbb{R}_+}(x_n)$, where $N_{\mathbb{R}_+}(0) = (-\infty, 0]$ and for all $x_i > 0$, $N_{\mathbb{R}_+}(x_i) = \{0\}$. Thus, (6.8) can also be formulated as

$$0 \in \nabla f(x) + \partial h(x) - z, \quad Xz = 0, \quad x \geq 0, \quad z \geq 0, \quad (6.9)$$

where $X = \text{diag}(x)$ and $Z = \text{diag}(z)$.

For fixed $x \in \mathbb{R}_+^n$ and $z \in \mathbb{R}_+^n$, we define approximations

$$\varphi^{\mathcal{L}}(s; x, z) := f(x) + (\nabla f(x) - z)^T s, \quad (6.10a)$$

$$\psi(s; x) \approx h(x + s) \text{ with } \psi(0; x) = h(x) \text{ and } \partial\psi(0; x) = \partial h(x), \quad (6.10b)$$

$$\hat{\psi}(s; x) := \psi(s; x) + \chi(x + s \mid \mathbb{R}_+^n), \quad (6.10c)$$

and the model of $f + h$ about x

$$m^{\mathcal{L}}(s; x, z, \nu) := \varphi^{\mathcal{L}}(s; x, z) + \frac{1}{2}\nu^{-1}\|s\|^2 + \psi(s; x), \quad (6.11)$$

where $\nu > 0$. We point out that $\nabla\varphi^{\mathcal{L}}(s; x, z) = \nabla f(x) - z$, which is the expression of the Lagrangian in the smooth case, thus, we use the superscript $\mathcal{L}$ to denote objects sharing similarities with the smooth Lagrangian.

For $\Delta \geq 0$, we further define

$$p^{\mathcal{L}}(\Delta; x, z, \nu) := \min_s \varphi^{\mathcal{L}}(s; x, z) + \frac{1}{2}\nu^{-1}\|s\|^2 + \hat{\psi}(s; x) + \chi(s \mid \Delta\mathbb{B}), \quad (6.12a)$$

$$P^{\mathcal{L}}(\Delta; x, z, \nu) := \underset{s}{\text{argmin}} \varphi^{\mathcal{L}}(s; x, z) + \frac{1}{2}\nu^{-1}\|s\|^2 + \hat{\psi}(s; x) + \chi(s \mid \Delta\mathbb{B}). \quad (6.12b)$$

Our associated optimality measure is

$$\xi^{\mathcal{L}}(\Delta; x, z, \nu) := f(x) + h(x) - \varphi^{\mathcal{L}}(s^{\mathcal{L}}; x, z) - \psi(s^{\mathcal{L}}; x), \quad (6.13)$$

where $s^{\mathcal{L}} \in P^{\mathcal{L}}(\Delta, x, z, \nu)$.

Lemma 9. *Let the CQ hold at $x \in \mathbb{R}_+^n$, $z \geq 0$ such that $Xz = 0$ and $\Delta > 0$. Then,*

$\xi^{\mathcal{L}}(\Delta; x, z, \nu) = 0 \iff 0 \in P^{\mathcal{L}}(\Delta; x, z, \nu) \implies x$ is first-order stationary for (6.1).

Proof. The first equivalence follows directly from (6.12b)–(6.13). The first-order necessary conditions for (6.12a) then imply

$$\begin{aligned} 0 &\in \nabla \varphi^{\mathcal{L}}(0; x, z) + \partial(\hat{\psi}(\cdot; x) + \chi(\cdot \mid \Delta \mathbb{B}))(0) \\ &= \nabla f(x) - z + \partial(\psi(\cdot; x) + \chi(\cdot \mid (-x + \mathbb{R}_+^n)) + \chi(\cdot \mid \Delta \mathbb{B}))(0) \\ &= \nabla f(x) - z + \partial(\psi(\cdot; x) + \chi(\cdot \mid (-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}))(0). \end{aligned} \quad (6.14)$$

As $(-x + \mathbb{R}_+^n)$ and $\Delta \mathbb{B}$ are convex, so is $(-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}$. From this observation, we deduce that,

$$\begin{aligned} N_{(-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}}(0) &= \partial \chi(0 \mid (-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}) \\ &= \partial \chi(0 \mid (-x + \mathbb{R}_+^n)) + \partial \chi(0 \mid \Delta \mathbb{B}) \\ &= \partial \chi(x \mid \mathbb{R}_+^n) \\ &= N_{\mathbb{R}_+^n}(x). \end{aligned}$$

The CQ combined with the above equations indicate that there is no $v \in \partial^\infty h(x) = \partial^\infty \psi(0; x)$, $v \neq 0$ such that $-v \in N_{\mathbb{R}_+^n}(x) = N_{(-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}}(0)$, thus, [73, Corollary 10.9] leads to

$$\begin{aligned} \partial(\psi(\cdot; x) + \chi(\cdot \mid (-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}))(0) &\subset \partial \psi(0; x) + N_{(-x + \mathbb{R}_+^n) \cap \Delta \mathbb{B}}(0) \\ &= \partial h(x) + N_{\mathbb{R}_+^n}(x). \end{aligned} \quad (6.15)$$

By injecting (6.15) into (6.14), we obtain

$$0 \in \nabla f(x) - z + \partial h(x) + N_{\mathbb{R}_+^n}(x).$$

From the observation above (6.9) and the fact that $Xz = 0$, we deduce that for any $v \in N_{\mathbb{R}_+^n}(x)$, $v - z \in N_{\mathbb{R}_+^n}(x)$. Thus,

$$0 \in \nabla f(x) + \partial h(x) + N_{\mathbb{R}_+^n}(x).$$

□

If h is convex, the CQ is not required in Lemma 9 [73, Exercise 10.8].

6.4 Projected-directions methods

Let us briefly recall the proximal gradient method [61] used to solve

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad f(s) + \tilde{h}(s), \quad (6.16)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has Lipschitz-continuous gradient and $\tilde{h} : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ is proper and lower semi-continuous. The method generates iterates s_k such that

$$s_{k+1} \in \underset{\nu \tilde{h}}{\text{prox}}(s_k - \nu \nabla f(s_k)) = \underset{s}{\text{argmin}} f(s_k) + \nabla f(s_k)^T (s - s_k) + \frac{1}{2} \nu^{-1} \|s - s_k\|^2 + \tilde{h}(s), \quad (6.17)$$

where $\nu > 0$, which leads to the first-order stationarity conditions

$$0 \in s_{k+1} - s_k + \nu \nabla f(s_k) + \nu \partial \tilde{h}(s_{k+1}). \quad (6.18)$$

A first approach to solving (6.1), that we can reformulate as

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + h(x) + \chi(x \mid \mathbb{R}_+^n),$$

is to use projected-directions methods. A simple example of such methods consists in performing the identification $\tilde{h} = h + \chi(\cdot \mid \mathbb{R}_+^n)$ in (6.16), and using the proximal gradient method as in (6.17).

Aravkin *et al.*'s TR and R2 are other examples of algorithms that can solve (6.1) with a similar strategy. Replacing h by $\tilde{h}$ and $\psi(\cdot; x)$ by $\tilde{\psi}(\cdot; x) = \psi(\cdot; x) + \chi(\cdot \mid -x + \mathbb{R}_+^n)$ in all models of TR and R2 is sufficient to generalize these methods to (6.1), if a solution of

$$s_k \in \underset{\nu_k \tilde{\psi}(\cdot; x_k) + \chi(\cdot \mid \Delta_k \mathbb{B})}{\text{prox}} \quad (-\nu_k \nabla f(x_k)). \quad (6.19)$$

for TR, or

$$s_k \in \underset{\nu_k \tilde{\psi}(\cdot; x_k)}{\text{prox}} \quad (-\nu_k \nabla f(x_k)) \quad (6.20)$$

for R2, is available.

Leconte and Orban [54] implement a variant of TR named TRDH that handles bound constraints for separable regularizers h (assuming that ψ is also separable). TRDH solves at each iteration k and for all $i \in \{1, \dots, n\}$ the problem

$$(s_k)_i \in \arg \min_{s_i} \nabla f(x_k)_i s_i + \frac{1}{2} (d_k)_i s_i^2 + (\psi(s; x_k))_i + \chi(s_i \mid \Delta_k \mathbb{B} \cap (-(x_k)_i + \mathbb{R}_+)), \quad (6.21)$$

with $(d_k)_i \in \mathbb{R}$. The special choice $(d_k)_i = \nu_k^{-1}$ shows that solving (6.21) for all i is equivalent to solving (6.19).

However, for nonseparable regularizers h , projected-directions methods rely on computing search directions such as (6.19), (6.20) or (6.21), which may be complicated (impossible for the latter), and therefore seems to be a limitation of this approach. The following section describes the implementation of a method that is different from projected directions methods, and is based upon interior-point techniques.

6.5 Barrier methods

Consider a sequence $\{\mu_k\} \searrow 0$.

Lemma 10. *Let ϕ_k be defined as in (6.3). Then, $\text{e-lim } \phi_k = \chi(\cdot \mid \mathbb{R}_+^n)$.*

Proof. It is sufficient to show that $\text{e-lim } \phi_{k,i} = \chi(\cdot \mid \mathbb{R}_+)$ for $i = 1, \dots, n$. Our goal is to bound each ϕ_k by two functions having $\chi(\cdot \mid \mathbb{R}_+^n)$ as epi-limit. We define

$$\phi_{k,i}^>(x) := \begin{cases} +\infty & \text{if } x \leq 0 \\ \phi_{k,i}(x) & \text{if } 0 < x < 1 \\ 0 & \text{if } x \geq 1, \end{cases} \quad \phi_{k,i}^<(x) := \begin{cases} +\infty & \text{if } x \leq 0 \\ 0 & \text{if } 0 < x < 1 \\ \phi_{k,i}(x) & \text{if } x \geq 1. \end{cases}$$

By construction, $\phi_{k,i}(x) = \phi_{k,i}^>(x) + \phi_{k,i}^<(x)$, $\{\phi_{k,i}^>(x)\} \searrow 0$ and $\{\phi_{k,i}^<(x)\} \uparrow 0$ as $k \rightarrow \infty$ for all $x > 0$. In particular, $\{\phi_{k,i}^>\} \xrightarrow{\text{p}} \chi(\cdot \mid \mathbb{R}_+)$ and $\{\phi_{k,i}^<\} \xrightarrow{\text{p}} \chi(\cdot \mid \mathbb{R}_+)$ as $k \rightarrow \infty$.

By [73, Proposition 7.4c], because $\{\phi_{k,i}^>\}$ is nonincreasing with k , its epi-limit is well defined and $\{\phi_{k,i}^>\} \xrightarrow{\text{e}} \text{cl inf}_k \phi_{k,i}^> = \chi(\cdot \mid \mathbb{R}_+)$.

Similarly, by [73, Proposition 7.4d], because $\{\phi_{k,i}^<\}$ is nondecreasing with k , its epi-limit is well defined and $\{\phi_{k,i}^<\} \xrightarrow{\text{e}} \text{sup}_k \text{cl } \phi_{k,i}^< = \chi(\cdot \mid \mathbb{R}_+)$.

Because $\phi_{k,i}^< \leq \phi_{k,i} \leq \phi_{k,i}^>$, [73, Proposition 7.4g] implies that $\{\phi_{k,i}\} \xrightarrow{\text{e}} \chi(\cdot \mid \mathbb{R}_+)$, and consequently, we obtain $\{\phi_k\} \xrightarrow{\text{e}} \chi(\cdot \mid \mathbb{R}_+^n)$. $\square$

Theorem 4. $\text{e-lim } f + h + \phi_k = f + h + \chi(\cdot \mid \mathbb{R}_+^n)$.

Proof. Lemma 10 and [73, Theorem 7.46a] imply $\{h + \phi_k\} \xrightarrow{\text{e}} h + \chi(\cdot \mid \mathbb{R}_+^n)$. Finally, because f is continuous, [73, Exercise 7.8a] yields $\{f + h + \phi_k\} \xrightarrow{\text{e}} f + h + \chi(\cdot \mid \mathbb{R}_+^n)$. $\square$

The following corollary legitimizes the barrier approach for (6.1).

Corollary 1. *Let $\inf\{f(x) + h(x) \mid x \geq 0\}$ be finite. For all $\epsilon \geq 0$,*

$$\limsup_k (\epsilon\text{-argmin } f + h + \phi_k) \subset \epsilon\text{-argmin } f + h + \chi(\cdot \mid \mathbb{R}_+^n).$$

In particular, if $\{\epsilon_k\} \searrow 0$,

$$\limsup_k (\epsilon_k\text{-argmin } f + h + \phi_k) \subset \text{argmin } f + h + \chi(\cdot \mid \mathbb{R}_+^n).$$

Proof. Follows directly from [73, Theorem 7.31b]. □

If $\inf\{f(x) + h(x) \mid x \geq 0\}$ is finite, the definition of the limit superior of a sequence of sets and the second part of Corollary 1 indicate that for any $\{\epsilon_k\} \searrow 0$, there exists $N \in \mathcal{N}_\infty^\#$ and $\bar{x}_k \in \epsilon_k\text{-argmin } f + h + \phi_k$ for all $k \in N$ such that $\{\bar{x}_k\}_N$ converges to a solution of (6.1).

6.5.1 Barrier subproblem

The k -th subproblem is

$$\underset{x}{\text{minimize}} \quad f(x) + h(x) + \phi_k(x). \quad (6.22)$$

$x_k^* \in \mathbb{R}_{++}^n$ is first-order stationary for (6.22) if

$$0 \in \nabla f(x_k^*) - \mu_k (X_k^*)^{-1} e + \partial h(x_k^*). \quad (6.23)$$

We call the process of solving (6.22) the k -th sequence of *inner* iterations, and we denote its iterates $x_{k,j}$ for $j \geq 0$. The definition of (6.22) along with certain parameter updates will be called an *outer* iteration.

For $x \in \mathbb{R}_{++}^n$ and $\delta \in (0, 1)$, let

$$\mathbb{R}_\delta^n(x) := \{s \in \mathbb{R}^n \mid \min_i (x + s)_i \geq \delta \min_i x_i\} \subset (-x + \mathbb{R}_{++}^n). \quad (6.24)$$

Note that $\mathbb{R}_\delta^n(x)$ is closed, and also convex, as shown in the following lemma.

Lemma 11. *Let $\delta \in (0, 1)$ and $x \in \mathbb{R}_{++}^n$. Then $\mathbb{R}_\delta^n(x)$ defined in (6.24) is convex.*

Proof. Let s_1 and $s_2 \in \mathbb{R}_\delta^n(x)$, and $t \in [0, 1]$. By definition, $\min_i (s_1 + x)_i \geq \delta \min_i x_i$ and

$\min_i (s_2 + x)_i \geq \delta \min_i x_i$. Now,

$$\begin{aligned}
\min_i (ts_1 + (1-t)s_2 + x)_i &\geq \min_i (t(s_1 + x))_i + \min_i ((1-t)(s_2 + x))_i \\
&= t \min_i (s_1 + x)_i + (1-t) \min_i (s_2 + x)_i \\
&\geq t\delta \min_i x_i + (1-t)\delta \min_i x_i \\
&= \delta \min_i x_i.
\end{aligned}$$

Thus, $ts_1 + (1-t)s_2 \in \mathbb{R}_\delta^n(x)$ and $\mathbb{R}_\delta^n(x)$ is convex. $\square$

Under the assumptions of Lemma 11 and for $\Delta > 0$, $\Delta\mathbb{B} \cap \mathbb{R}_\delta^n(x)$ is convex.

At outer iteration k , we choose $\delta_k \in (0, 1)$, and solve (6.22) inexactly by approximately solving a sequence of trust-region subproblems of the form

$$\underset{s}{\text{minimize}} \ m(s; x_{k,j}) + \chi(s \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})), \quad (6.25a)$$

$$m(s; x_{k,j}) := \varphi(s; x_{k,j}) + \psi(s; x_{k,j}), \quad (6.25b)$$

where $\varphi(s; x_{k,j}) \approx (f + \phi_k)(x_{k,j} + s)$ and $\psi(s; x_{k,j}) \approx h(x_{k,j} + s)$ model the smooth and nonsmooth parts of (6.22), respectively, and $\Delta_{k,j} > 0$ is a trust-region radius. Models are required to satisfy the following assumption.

Model Assumption 6.5.1. *For any $k \in \mathbb{N}$ and $j \in \mathbb{N}$, $\varphi(\cdot; x_{k,j})$ is continuously differentiable on $\mathbb{R}_{++}^n$ with $\varphi(0; x_{k,j}) = f(x_{k,j}) + \phi_k(x_{k,j})$ and $\nabla_s \varphi(0; x_{k,j}) = \nabla f(x_{k,j}) + \nabla \phi_k(x_{k,j})$. In addition, $\nabla_s \varphi(\cdot; x_{k,j})$ is Lipschitz continuous with constant $L_{k,j} \geq 0$. We require that $\psi(\cdot; x_{k,j})$ be proper, lsc, and satisfy $\psi(0; x_{k,j}) = h(x_{k,j})$ and $\partial\psi(0; x_{k,j}) = \partial h(x_{k,j})$.*

Proposition 22. *Let Model Assumption 6.5.1 be satisfied. Then $s = 0$ is first-order stationary for (6.25) if and only if $x_{k,j}$ is first-order stationary for (6.22).*

Proof. If $x_{k,j}$ is first-order stationary, then $0 \in \nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1}e + \partial h(x_{k,j}) = \nabla_s \varphi(0; x_{k,j}) + \partial\psi(0; x_{k,j})$. Note that $x_{k,j} > 0$ so $\partial\chi(0 \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = \{0\}$, and $\Delta_{k,j} > 0$, so there is an open set $O \subset \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})$ such that for all $s \in O$, $\chi(s \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = 0$. Thus, using the definition of the subdifferential, $\partial(\psi(\cdot; x_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))) (0) = \partial\psi(0; x_{k,j})$. We conclude that $0 \in \nabla_s \varphi(0; x_{k,j}) + \partial(\psi(\cdot; x_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))) (0)$, which is the definition of $s = 0$ being first-order stationary for (6.25). The reciprocal can also be established from these observations, because if $0 \in \nabla_s \varphi(0; x_{k,j}) + \partial(\psi(\cdot; x_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))) (0)$, we have shown that $0 \in \nabla_s \varphi(0; x_{k,j}) + \partial\psi(0; x_{k,j}) = \nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1}e + \partial h(x_{k,j})$. $\square$

As a special case of Proposition 22, if $s = 0$ solves (6.25), then $x_{k,j}$ is first-order stationary for (6.22).

Let

$$\varphi_f(s; x_{k,j}, B_{k,j}) := f(x_{k,j}) + \nabla f(x_{k,j})^T s + \frac{1}{2} s^T B_{k,j} s, \quad (6.26)$$

where $B_{k,j} = B_{k,j}^T$, be a second order Taylor approximation of f about $x_{k,j}$. We are particularly interested in the quadratic model

$$\begin{aligned} \varphi(s; x_{k,j}, B_{k,j}) &:= \varphi_f(s; x_{k,j}, B_{k,j}) + \phi_k(x_{k,j}) - \mu_k e^T X_{k,j}^{-1} s + \frac{1}{2} s^T X_{k,j}^{-1} Z_{k,j} s \\ &= (f + \phi_k)(x_{k,j}) + (\nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1} e)^T s + \frac{1}{2} s^T (B_{k,j} + X_{k,j}^{-1} Z_{k,j}) s, \end{aligned} \quad (6.27)$$

where $z_{k,j}$ is an approximation to the vector of multipliers for the bound constraints of (6.1).

Let $s_{k,j}$ be an approximate solution of (6.25). If $s_{k,j}$ is accepted as a step for our algorithm used to solve (6.22) (the acceptance condition is detailed in Algorithm 6.5.2), we perform the update $x_{k,j+1} = x_{k,j} + s_{k,j}$.

By analogy with the smooth case, we use $z_{k,j} := \mu_k X_{k,j}^{-1} e$ when $x_{k,j}$ is first-order stationary for (6.22). Multiplying through by $X_{k,j}$, we obtain $X_{k,j} z_{k,j} = \mu_k e$. Linearizing the continuous equality $Xz - \mu e = 0$ with respect to x and z and evaluating all quantities at iteration (k, j) yields $X_{k,j} \Delta z_{k,j} + Z_{k,j} s_{k,j} = \mu_k e - X_{k,j} z_{k,j}$, which suggests that if $x_{k,j+1} = x_{k,j} + s_{k,j}$, then $z_{k,j+1} = z_{k,j} + \Delta z_{k,j} = \mu_k X_{k,j}^{-1} e - X_{k,j}^{-1} Z_{k,j} s_{k,j}$.

However, the latter $z_{k,j+1}$ may not be positive. We perform the update described by Conn *et al.* [34], by defining

$$\hat{z}_{k,j+1} = \mu_k X_{k,j}^{-1} e - X_{k,j}^{-1} Z_{k,j} s_{k,j}, \quad (6.28)$$

and projecting $\hat{z}_{k,j+1}$ componentwise into the following interval to get $z_{k,j+1}$

$$\mathcal{I} = [\kappa_{\text{zul}} \min(e, z_{k,j}, \mu_k X_{k,j+1}^{-1} e), \max(\kappa_{\text{zuu}} e, z_{k,j}, \kappa_{\text{zuu}} \mu_k^{-1} e, \kappa_{\text{zuu}} \mu_k X_{k,j+1}^{-1} e)], \quad (6.29)$$

with $0 < \kappa_{\text{zul}} < 1 < \kappa_{\text{zuu}}$. Projecting $\hat{z}_{k,j+1}$ into (6.29) always generates a positive $z_{k,j+1}$. The choice $z_{k,j+1} = z_{k,j}$ is also available. The other bounds of (6.29) will be useful in Section 6.5.2 and Section 6.5.3.

We define the following model, based upon a first-order Taylor approximation

$$\varphi_{\text{cp}}(s; x_{k,j}) := (f + \phi_k)(x_{k,j}) + (\nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1} e)^T s, \quad (6.30a)$$

$$m_{\text{cp}}(s; x_{k,j}, \nu_{k,j}) := \varphi_{\text{cp}}(s; x_{k,j}) + \frac{1}{2} \nu_{k,j}^{-1} \|s\|^2 + \psi(s; x_{k,j}), \quad (6.30b)$$

where “cp” stands for “Cauchy point”. Let $s_{k,j,1}$ be the solution of (6.25) with model

$m_{\text{cp}}(s; x_{k,j}, \nu_{k,j})$. As stated in [4, Section 3.2], $s_{k,j,1}$ is actually the first step of the proximal gradient method (6.17) from $s_{k,j,0} = 0$ applied to the minimization of $\varphi_{\text{cp}} + \psi$ with step length $\nu_{k,j}$:

$$s_{k,j,1} \in \underset{\nu_{k,j}\psi(\cdot; x_{k,j}) + \chi(\cdot | \Delta_{k,j} \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))}{\text{prox}} (-\nu_{k,j} \nabla \varphi_{\text{cp}}(0; x_{k,j}, \nu_{k,j})). \quad (6.31)$$

Let

$$\xi_{\text{cp}}(\Delta; x_{k,j}, \nu_{k,j}) := (f + \phi_k + h)(x_{k,j}) - (\varphi_{\text{cp}} + \psi)(s_{k,j,1}; x_{k,j}), \quad (6.32)$$

where $\Delta > 0$, and let $\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta; x_{k,j}, \nu_{k,j})^{1/2}$ be our measure of criticality. Aravkin *et al.* [4] and its corrigendum [3] indicate that $\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta; x_{k,j}, \nu_{k,j})^{1/2}$ is similar to $\nu_{k,j}^{-1} \|s_{k,j,1}\|$, which is the norm of the generalized gradient at $x_{k,j}$. We can apply [73, Theorems 1.17 and 7.41] to conclude that $\xi_{\text{cp}}(\Delta; x_{k,j}, \nu_{k,j})$ is proper lsc in $(x_{k,j}, \nu_{k,j}) \in \mathbb{R}_{++}^n \times \mathbb{R}_{++}$. In particular, $\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta; x_{k,j}, \nu_{k,j})^{1/2} = 0$ for any $\Delta > 0$ and $\nu_{k,j} > 0 \implies s = 0$ solves (6.25), and $x_{k,j}$ is first-order stationary for (6.22).

Algorithm 6.5.1 summarizes the outer iteration.

Algorithm 6.5.1 Nonsmooth interior-point method (outer iteration).

- 1: Choose $\epsilon > 0$, sequences $\{\mu_k\} \searrow 0$, $\{\epsilon_{d,k}\} \searrow 0$, $\{\epsilon_{p,k}\} \searrow 0$, and $\{\delta_k\} \rightarrow \bar{\delta} \in [0, 1)$ with $\delta_k \in (0, 1)$ for all k .
- 2: Choose $x_{0,0} \in \mathbb{R}_{++}^n$ where h is finite.
- 3: **for** $k = 0, 1, \dots$ **do**
- 4: Compute an approximate solution $x_k := x_{k,j}$ to (6.22) and $z_k := z_{k,j}$ in the sense that

$$\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} \leq \epsilon_{d,k} \quad (6.33)$$

and

$$\|X_{k,j} z_{k,j} - \mu_k e\| \leq \epsilon_{p,k}. \quad (6.34)$$

- 5: Set $x_{k+1,0} := x_k$.
-

For each outer iteration k , the inner iterations generate a sequence $\{x_{k,j}\}$ according to an adaptation of [4, Algorithm 3.1] in which the subproblems have the form (6.25) with the smooth part of the model defined by (6.27). Each trust-region step is required to satisfy the following assumption.

Step Assumption 6.5.1. *Let $k \in \mathbb{N}$. There exists $\kappa_{\text{m},k} > 0$ and $\kappa_{\text{mdc},k} \in (0, 1)$ such that for all j , $s_{k,j} \in \Delta_{k,j} \mathbb{B} \cap \mathbb{R}_{\delta_k}(x_{k,j})$,*

$$|(f + \phi_k + h)(x_{k,j} + s_{k,j}) - m(s_{k,j}; x_{k,j}, B_{k,j})| \leq \kappa_{\text{m},k} \|s_{k,j}\|^2, \quad (6.35a)$$

$$m(0; x_{k,j}, B_{k,j}) - m(s_{k,j}; x_{k,j}, B_{k,j}) \geq \kappa_{\text{mdc},k} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j}), \quad (6.35b)$$

where m is defined in (6.25b)–(6.27), and $\xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})$ is defined in (6.32).

In Step Assumption 6.5.1, the subscript “m” of $\kappa_{\text{m},k}$ refers to the model adequacy, and the subscript “mdc” of $\kappa_{\text{mdc},k}$ refers to the model decrease.

Algorithm 6.5.2 summarizes the process.

Algorithm 6.5.2 Nonsmooth interior-point method (inner iteration).

1: Choose constants

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 \leq \gamma_2 < 1 < \gamma_3 \leq \gamma_4, \quad \Delta_{\max} > \Delta_{k,0} > 0, \quad \alpha > 0, \beta \geq 1.$$

2: Compute $f(x_{k,0}) + \phi_k(x_{k,0}) + h(x_{k,0})$.

3: **for** $j = 0, 1, \dots$ **do**

4: Choose $0 < \nu_{k,j} \leq 1/(L_{k,j} + \alpha^{-1}\Delta_{k,j}^{-1})$.

5: Define $m(s; x_{k,j}, B_{k,j})$ as in (6.25b) satisfying Model Assumption 6.5.1.

6: Define $m_{\text{cp}}(s; x_{k,j}, \nu_{k,j})$ as in (6.30b).

7: Compute $s_{k,j,1}$ as a solution of (6.31).

8: Compute an approximate solution $s_{k,j}$ of (6.25) such that $\|s_{k,j}\| \leq \min(\Delta_{k,j}, \beta\|s_{k,j,1}\|)$.

9: Compute the ratio

$$\rho_{k,j} := \frac{(f + \phi_k + h)(x_{k,j}) - (f + \phi_k + h)(x_{k,j} + s_{k,j})}{m(0; x_{k,j}, B_{k,j}) - m(s_{k,j}; x_{k,j}, B_{k,j})}.$$

10: If $\rho_{k,j} \geq \eta_1$, set $x_{k,j+1} = x_{k,j} + s_{k,j}$ and update $z_{k,j+1}$ according to (6.28) and (6.29).
 Otherwise, set $x_{k,j+1} = x_{k,j}$ and $z_{k,j+1} = z_{k,j}$.

11: Update the trust-region radius according to

$$\hat{\Delta}_{k,j+1} \in \begin{cases} [\gamma_3\Delta_{k,j}, \gamma_4\Delta_{k,j}] & \text{if } \rho_{k,j} \geq \eta_2, & \text{(very successful iteration)} \\ [\gamma_2\Delta_{k,j}, \Delta_{k,j}] & \text{if } \eta_1 \leq \rho_{k,j} < \eta_2, & \text{(successful iteration)} \\ [\gamma_1\Delta_{k,j}, \gamma_2\Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1 & \text{(unsuccessful iteration)} \end{cases}$$

$$\text{and } \Delta_{k,j+1} = \min(\hat{\Delta}_{k,j+1}, \Delta_{\max}).$$

6.5.2 Convergence of the inner iterations

Let k and j be fixed positive integers, and $x_{k,j} > 0$. We may rewrite our “Cauchy point” subproblem for the inner iterations as in [4]:

$$p(\Delta; x_{k,j}, \nu_{k,j}, \delta_k) := \underset{s}{\text{minimize}} m_{\text{cp}}(s; x_{k,j}, \nu_{k,j}) + \chi(s; \Delta\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) \quad (6.36a)$$

$$P(\Delta; x_{k,j}, \nu_{k,j}, \delta_k) := \underset{s}{\text{argmin}} m_{\text{cp}}(s; x_{k,j}, \nu_{k,j}) + \chi(s; \Delta\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})). \quad (6.36b)$$

First, we present some properties of the subproblem (6.36a) in the following result.

Proposition 23 (4, Proposition 3.1). *Let Model Assumption 6.5.1 be satisfied, $\nu > 0$, $\delta > 0$ and $x \in \mathbb{R}_+^n$. If we define $p(0; x, \nu, \delta) := \varphi_{\text{cp}}(0; x) + \psi(0; x)$ and $P(0; x, \nu, \delta) = \{0\}$, the domain of $p(\cdot; x, \nu, \delta)$ and $P(\cdot; x, \nu, \delta)$ is $\{\Delta \mid \Delta \geq 0\}$. In addition,*

1. $p(\cdot; x, \nu, \delta)$ is proper lsc and for each $\Delta \geq 0$, $P(\Delta; x, \nu, \delta)$ is nonempty and compact;
2. if $\{\Delta_{k,j}\} \rightarrow \bar{\Delta}_k \geq 0$ in such a way that $\{p(\Delta_{k,j}; x, \nu, \delta)\} \rightarrow p(\bar{\Delta}_k; x, \nu, \delta)$, and for each j , $s_{k,j} \in P(\Delta_{k,j}; x, \nu, \delta)$, then $\{s_{k,j}\}$ is bounded and all its limit points are in $P(\bar{\Delta}_k; x, \nu, \delta)$;
3. if $\varphi_{\text{cp}}(\cdot; x) + \frac{1}{2}\nu^{-1}\|s\|^2 + \psi(\cdot; x)$ is strictly convex, $P(\Delta; x, \nu, \delta)$ is single valued;
4. if $\bar{\Delta}_k > 0$ and there exists $\bar{s} \in P(\bar{\Delta}_k; x, \nu, \delta)$ such that $\bar{s} \in \text{int}(\bar{\Delta}_k \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x))$, then $p(\cdot; x, \nu, \delta)$ is continuous at $\bar{\Delta}_k$ and $\{p(\Delta_{k,j}; x, \nu, \delta)\} \rightarrow p(\bar{\Delta}_k; x, \nu, \delta)$ holds in part 2.

Proof. Model Assumption 6.5.1 and the compactness of $\Delta \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x)$ ensure that the objective of (6.36a) is always level-bounded in s locally uniformly in Δ , because for any $\bar{\Delta} > 0$, $\epsilon > 0$, and $\Delta \in (\bar{\Delta} - \epsilon, \bar{\Delta} + \epsilon)$ with $\Delta \geq 0$, its level sets are contained in $\Delta \mathbb{B} \cap \mathbb{R}_{\delta_k}(x_{k,j}) \subseteq (\bar{\Delta} + \epsilon) \mathbb{B} \cap \mathbb{R}_{\delta_k}(x_{k,j})$. From this observation, we can draw similar conclusions to the analysis of [4, Proposition 3.1]. $\square$

The observation in the proof of Proposition 23 and Model Assumption 6.5.1 allows us to derive directly some of the convergence properties of Aravkin *et al.* [4] for Algorithm 6.5.2.

Proposition 24 (4, Theorem 3.4). *Let Model Assumption 6.5.1 and Step Assumption 6.5.1 be satisfied and let*

$$\Delta_{\text{succ},k} := \frac{\kappa_{\text{mdc},k}(1 - \eta_2)}{2\kappa_{\text{m},k}\alpha\beta^2} > 0. \quad (6.37)$$

If $x_{k,j}$ is not first-order stationary for (6.22) and $\Delta_{k,j} \leq \Delta_{\text{succ}}$, then iteration j is very successful and $\Delta_{k,j+1} \geq \Delta_{k,j}$.

Proof. If $x_{k,j}$ is not first-order stationary, $\mathbb{R}_{\delta_k}^n(x_{k,j}) \neq \{0\}$, thus $s_{k,j,1} \neq 0$ and $s_{k,j} \neq 0$. The rest of the proof is identical to that of [4, Theorem 3.4]. $\square$

Now, let $\Delta_{\min,k} := \min(\Delta_{k,0}, \gamma_1 \Delta_{\text{succ},k}) > 0$. Then, $\Delta_{k,j} \geq \Delta_{\min,k}$ for all $j \in \mathbb{N}$. If we consider φ defined in (6.27), for s_1 and s_2 in $\mathbb{R}_+^n$,

$$\|\nabla\varphi(s_1) - \nabla\varphi(s_2)\| = \|B_{k,j}(s_1 - s_2) + X_{k,j}^{-1}Z_{k,j}(s_1 - s_2)\|. \quad (6.38)$$

As $L_{k,j} = \|B_{k,j} + X_{k,j}^{-1}Z_{k,j}\| \leq \|B_{k,j}\| + \|X_{k,j}^{-1}\|\|Z_{k,j}\|$, if $\{B_{k,j}\}_j$ remains bounded, $\{x_{k,j}\}_j$ must be bounded away from zero to guarantee the existence of some $M_k > 0$ such that $L_{k,j} \leq M_k$ for all $j \in \mathbb{N}$.

To apply the complexity results, and to establish that $\liminf \nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} = 0$ if $f + h + \phi_k$ is bounded below on $\mathbb{R}_+^n$, we need a stronger assumption on the Lipschitz constant of our model.

Model Assumption 6.5.2 (4, Model Assumption 3.3). *In Model Assumption 6.5.1, there exists $M_k > 0$ such that $0 \leq L_{k,j} \leq M_k$ for all $j \in \mathbb{N}$. In addition, we select $\nu_{k,j}$ in line 4 of Algorithm 6.5.2 in a way that there exists $\nu_{\min,k} > 0$ such that $\nu_{k,j} \geq \nu_{\min,k}$ for all $j \in \mathbb{N}$.*

As in Aravkin *et al.* [4], we can set $\nu_{k,j} := 1/(L_{k,j} + \alpha^{-1} \Delta_{k,j}^{-1})$ in Algorithm 6.5.2 to ensure $\nu_{k,j} \geq \nu_{\min,k} := 1/(M_k + \alpha^{-1} \Delta_{\min,k}^{-1}) > 0$ if the first part of Model Assumption 6.5.2 holds with M_k . The observations below (6.38) motivate us to prove that $\{x_{k,j}\}_j$ is bounded away from zero in the next result.

Proposition 25. *Let $k \in \mathbb{N}$, Model Assumption 6.5.1 be satisfied for φ in (6.27), and $(f + h)(x_{k,j}) \geq (f + h)_{\text{low},k}$. Then, there exists $\kappa_{\text{mdb},k} > 0$ such that, for all j , we have*

$$\min_i (x_{k,j})_i \geq \kappa_{\text{mdb},k}. \quad (6.39)$$

Proof. We proceed similarly as in Conn *et al.* [34, Theorem 13.2.1]. Let k be a positive integer and $\{x_{k,j}\}$ be a sequence generated by Algorithm 6.5.2. As $\{(f + \phi_k + h)(x_{k,j})\}_j$ is decreasing and $(f + h)(x_{k,j}) \geq (f + h)_{\text{low},k}$, we have $\limsup_j \phi_k(x_{k,j}) < \infty$, which implies that (6.39) holds. $\square$

The following proposition shows that we can use the convergence results of Aravkin *et al.* [4] using the same assumptions they used for φ_f . It will justify that Model Assumption 6.5.2 can be used for φ defined in (6.27).

Proposition 26. *Under the assumptions of Proposition 25, let φ_f be defined as in (6.26) so that $\nabla_s \varphi_f(\cdot; x_{k,j}, B_{k,j})$ is Lipschitz continuous with constant $\tilde{L}_{k,j} \geq 0$ and there exists $\tilde{M}_k > 0$ such that $0 \leq \tilde{L}_{k,j} \leq \tilde{M}_k$ for all $j \in \mathbb{N}$. Then φ satisfies Model Assumption 6.5.2.*

Proof. We can use (6.29) and (6.39) to say that $X_{k,j}^{-1} Z_{k,j}$ is bounded for all j , and we deduce from (6.38) that φ satisfies Model Assumption 6.5.2. $\square$

Now, we justify that Step Assumption 6.5.1 holds when $h(x_{k,j} + s_{k,j}) = \psi(s_{k,j}; x_{k,j})$. As $\{x_{k,j}\}_j$ remains bounded away from $\partial \mathbb{R}_+^n$ with Proposition 25, so does $\{x_{k,j} + s_{k,j}\}_j$ by definition of $\mathbb{R}_{\delta_k}(x_{k,j})$. Since ∇f is Lipschitz-continuous,

$$f(x_{k,j} + s_{k,j}) - f(x_{k,j}) - \nabla f(x_{k,j})^T s_{k,j} \leq \frac{1}{2} L_f \|s_{k,j}\|^2, \quad (6.40)$$

and a second-order Taylor approximation of ϕ_k about $x_{k,j} + s_{k,j}$ gives

$$\phi_k(x_{k,j} + s_{k,j}) - \phi_k(x_{k,j}) - \mu_k s_{k,j}^T X_{k,j}^{-1} e = \frac{1}{2} \mu_k s_{k,j}^T X_{k,j}^{-2} s_{k,j} + o(\|s_{k,j}\|^2). \quad (6.41)$$

Under the assumptions of Proposition 26, $L_{k,j} = \|B_{k,j} + X_{k,j}^{-1} Z_{k,j}\| \leq M_k$, and

$$\begin{aligned} & |(f + \phi_k + h)(x_{k,j} + s_{k,j}) - m(s_{k,j}; x_{k,j}, B_{k,j})| \\ &= |f(x_{k,j} + s_{k,j}) - f(x_{k,j}) - \nabla f(x_{k,j})^T s_{k,j} + \phi_k(x_{k,j} + s_{k,j}) - \phi_k(x_{k,j}) - \\ & \quad \mu_k s_{k,j}^T X_{k,j}^{-1} e - \frac{1}{2} s_{k,j}^T (B_{k,j} + X_{k,j}^{-1} Z_{k,j}) s_{k,j}| + o(\|s_{k,j}\|^2). \end{aligned}$$

The above equality combined with (6.40) and (6.41) implies that (6.35a) holds.

To emphasize the similarities between our inner iterations and the trust-region algorithm of Aravkin *et al.* [4], and in light of Proposition 26, we use in our next results that φ satisfies Model Assumption 6.5.2, instead of writing assumptions on φ_f . The following proposition gives us a sufficient condition for (6.35b) to be satisfied.

Proposition 27 (3, Proposition 1). *If Model Assumption 6.5.2 is satisfied with bounded Hessian approximations $\{B_{k,j}\}_j$, then there exists $\kappa_{\text{mdc},k} \in (0, 1)$ such that (6.35b) holds for all j .*

Proof. The proof is identical to that of [3, Proposition 1] when replacing B_k by $B_{k,j} + X_{k,j}^{-1} Z_{k,j}$, and using the subscripts $_{k,j}$ where j is the iteration number of the algorithm instead of the subscript $_k$. $\square$

Proposition 25 allows us to write the following convergence results for Algorithm 6.5.2. As in Aravkin *et al.* [4], we define the smallest iteration number $j_k(\epsilon)$ such that

$$\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} \leq \epsilon \quad (0 < \epsilon < 1), \quad k = 0, 1, \dots, \quad (6.42)$$

and we express the set of successful iterations, the set of successful iterations for which (6.42) has not yet been attained, and the set of unsuccessful iterations for which (6.42) has not yet been attained as

$$\mathcal{S}_k := \{j \in \mathbb{N} \mid \rho_{k,j} \geq \eta_1\} \quad (6.43a)$$

$$\mathcal{S}_k(\epsilon) := \{j \in \mathcal{S}_k \mid j < j_k(\epsilon)\} \quad (6.43b)$$

$$\mathcal{U}_k(\epsilon) := \{j \in \mathbb{N} \mid j \notin \mathcal{S}_k \text{ and } j < j_k(\epsilon)\}. \quad (6.43c)$$

Proposition 28 (4, Theorem 3.5). *Let Model Assumption 6.5.1 and Step Assumption 6.5.1 be satisfied. If Algorithm 6.5.2 only generates finitely many successful iterations, then $x_{k,j} = x_k^*$ for all sufficiently large j and x_k^* is first-order critical for (6.22).*

Proof. The proof is inspired from [4, Theorem 3.5], which itself follows that of Conn *et al.* [34, Theorem 6.6.4]. The assumptions indicate that there is $j_0 \in \mathbb{N}$ such that all iterations $j \geq j_0$ are unsuccessful, and $x_{k,j} = x_{k,j_0} = x_k^*$ because of the update rules of Algorithm 6.5.2. We assume by contradiction that x_k^* is not first-order critical. x_k^* does not have any of its components equal to $+\infty$ because it is attained after a finite number of iterations of Algorithm 6.5.2. As h is proper, $(f + h)(x_k^*) > -\infty$. Thus, Proposition 25 implies that there exists Δ_k^* such that $\Delta_k^* \mathbb{B} \subset \mathbb{R}_{\delta_k}(x_k^*)$. Since all iterations $j \geq j_0$ are unsuccessful, there will be some $j_1 \geq j_0$ such that $\Delta_{j_1} \leq \min(\Delta_{\text{succ},k}, \Delta_k^*)$, which implies that iteration j_1 is very successful with Proposition 24, and contradicts the fact that x_k^* is not first-order critical. $\square$

Finally, we have the following result for the inner iterates.

Proposition 29 (4, Theorem 3.11). *Let $k \in \mathbb{N}$, Model Assumption 6.5.1 and Model Assumption 6.5.2 be verified for φ defined in (6.27), and Step Assumption 6.5.1 be satisfied for Algorithm 6.5.2. If there are infinitely many successful iterations, then, either*

$$\lim_{j \rightarrow \infty} (f + \phi_k + h)(x_{k,j}) = -\infty \quad \text{or} \quad \lim_{j \rightarrow \infty} \nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} = 0. \quad (6.44)$$

Proof. The proof is identical as that of [4, Theorem 3.11]. $\square$

Now, the definitions of φ_{cp} in (6.30a) and $s_{k,j,1}$ as a minimizer of (6.25) with model m_{cp} indicate that

$$\begin{aligned} (f + \phi_k + h)(x_{k,j}) &= \varphi_{\text{cp}}(0; x_{k,j}) + \psi(0; x_{k,j}) = m_{\text{cp}}(0; x_{k,j}, \nu_{k,j}) \geq \\ &m_{\text{cp}}(s_{k,j,1}; x_{k,j}, \nu_{k,j}) = \varphi_{\text{cp}}(s_{k,j,1}; x_{k,j}) + \frac{1}{2} \nu_{k,j}^{-1} \|s_{k,j,1}\|^2 + \psi(s_{k,j,1}; x_{k,j}). \end{aligned}$$

By reinjecting this inequality into the definition of ξ_{cp} in (6.32), we obtain

$$\xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j}) \geq \frac{1}{2} \nu_{k,j}^{-1} \|s_{k,j,1}\|^2,$$

so that

$$\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} \geq \frac{1}{\sqrt{2}} \nu_{k,j}^{-1} \|s_{k,j,1}\|. \quad (6.45)$$

From this observation, we deduce that $s_{k,j} \xrightarrow{j \rightarrow \infty} 0$ in the following result.

Lemma 12. *If Model Assumption 6.5.1 holds for φ defined in (6.27) and*

$$\lim_{j \rightarrow \infty} \nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} \rightarrow 0,$$

then $\lim_{j \rightarrow \infty} \|s_{k,j,1}\| = \lim_{j \rightarrow \infty} \|s_{k,j}\| = 0$.

Proof. We use $\beta \|s_{k,j,1}\|^2 \geq \|s_{k,j}\|^2$ and (6.45) to conclude that

$$\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} \geq \frac{1}{\sqrt{2}} \nu_{k,j}^{-1} \|s_{k,j,1}\| \geq \frac{1}{\sqrt{2}} \nu_{k,j}^{-1} \beta^{-1/2} \|s_{k,j}\| \geq 0.$$

With Model Assumption 6.5.1, $\nu_{k,j} \xrightarrow{j \rightarrow \infty} \bar{\nu}_k > 0$, and we have $\|s_{k,j}\| \xrightarrow{j \rightarrow \infty} 0$. $\square$

Now, we study the asymptotic satisfaction of the inner perturbed complementarity. We show that (6.34) is eventually satisfied, similarly to [34, Theorem 13.6.4] in the smooth case.

Proposition 30. *Let $k \in \mathbb{N}$, Model Assumption 6.5.1 and Model Assumption 6.5.2 be verified for φ defined in (6.27), Step Assumption 6.5.1 be satisfied for Algorithm 6.5.2, and $(f + \phi_k + h)(x_{k,j}) \geq (f + \phi_k + h)_{\text{low},k}$ for all $j \in \mathbb{N}$. Then,*

$$\lim_{j \rightarrow \infty} \|\mu_k X_{k,j}^{-1} e - z_{k,j}\| = 0.$$

Proof. We proceed similarly as in the proof of [34, Theorem 13.6.4]. With the formula

$$\hat{z}_{k,j+1} = \mu_k X_{k,j}^{-1} e - X_{k,j}^{-1} Z_{k,j} s_{k,j},$$

$$\begin{aligned} \|\hat{z}_{k,j+1} - \mu_k X_{k,j+1}^{-1} e\| &\leq \|X_{k,j}^{-1} Z_{k,j} s_{k,j}\| + \mu_k \|X_{k,j+1}^{-1} e - X_{k,j}^{-1} e\| \\ &\leq \|X_{k,j}^{-1} Z_{k,j}\| \|s_{k,j}\| + \mu_k \sqrt{n} \|X_{k,j+1}^{-1} - X_{k,j}^{-1}\|. \end{aligned}$$

Using Proposition 29, we have $\lim_{j \rightarrow \infty} \nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} = 0$. This leads to

$$\lim_{j \rightarrow \infty} \|s_{k,j}\| = 0$$

with Lemma 12, and we have that $x_{k,j}$ is bounded away from 0 using Proposition 25. Therefore, either iteration j is not successful and $X_{k,j+1} = X_{k,j}$, or

$$\begin{aligned} \|X_{k,j+1}^{-1} - X_{k,j}^{-1}\| &= \|X_{k,j}^{-1} (X_{k,j} X_{k,j+1}^{-1} - I)\| \\ &= \|X_{k,j}^{-1} (X_{k,j} (X_{k,j} + S_{k,j})^{-1} - I)\| \\ &= \|X_{k,j}^{-1} ((I + X_{k,j}^{-1} S_{k,j})^{-1} - I)\| \rightarrow 0. \end{aligned}$$

Since $X_{k,j+1}^{-1}z_{k,j}$ is also bounded for all j using Proposition 25 and (6.29), we have

$$\lim_{j \rightarrow \infty} \|\hat{z}_{k,j+1} - \mu_k X_{k,j+1}^{-1}e\| \rightarrow 0. \quad (6.46)$$

For j large enough, we have

$$\kappa_{\text{zul}}\mu_k X_{k,j+1}^{-1}e \leq \hat{z}_{k,j+1} \leq \kappa_{\text{uu}}\mu_k X_{k,j+1}^{-1}e, \quad (6.47)$$

so that $\hat{z}_{k,j+1} = z_{k,j+1}$ if j is large enough. $\square$

In Algorithm 6.5.1, each subproblem is solved approximately with tolerances $\epsilon_{d,k} \searrow 0$ and $\epsilon_{p,k} \searrow 0$. This gives rise to the analysis in Section 6.5.3.

6.5.3 Convergence of the outer iterations

For each $k \in \mathbb{N}$, the stopping condition of Algorithm 6.5.2 occurs in a finite number of iterations. Let j_k denote the number of iterations performed by Algorithm 6.5.2 at outer iteration k . To simplify the notation, let $\bar{x}_k := x_{k,j_k}$, $\bar{z}_k := z_{k,j_k}$, $\bar{s}_{k,1} := s_{k,j_k,1}$, $\bar{\Delta}_k := \Delta_{k,j_k}$ and $\bar{\nu}_k := \nu_{k,j_k}$.

First, we present two assumptions that will be useful for our analysis.

Parameter Assumption 6.5.1. $\liminf \Delta_{\min,k} = \bar{\Delta} > 0$ and $\liminf \bar{\nu}_k = \bar{\nu} > 0$.

To satisfy the second part of Parameter Assumption 6.5.1, we need to change φ in (6.27) to

$$\begin{aligned} \varphi(s; x_{k,j}, B_{k,j}) &:= \varphi_f(s; x_{k,j}, B_{k,j}) + \phi_k(x_{k,j}) - \mu_k e^T X_{k,j}^{-1}s + \frac{1}{2}s^T \Theta_{k,j}s \\ &= (f + \phi_k)(x_{k,j}) + (\nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1}e)^T s + \frac{1}{2}s^T (B_{k,j} + \Theta_{k,j})s, \end{aligned} \quad (6.48)$$

where $\Theta_{k,j} = \min(X_{k,j}^{-1}Z_{k,j}, \kappa_{\text{bar}}I)$ with the min taken componentwise and $\kappa_{\text{bar}} > 0$.

Since the results of Section 6.5.2 involving φ defined in (6.27) are all based upon Step Assumption 6.5.1, those results continue to apply if Step Assumption 6.5.1 holds for φ defined in (6.48). We now show that that is the case.

First, we observe that

$$\begin{aligned} (f + \phi_k)(x_{k,j} + s_{k,j}) - \varphi(s_{k,j}; x_{k,j}, B_{k,j}) &= f(x_{k,j} + s_{k,j}) - f(x_{k,j}) - \nabla f(x_{k,j})^T s_{k,j} + \\ &\quad \phi_k(x_{k,j} + s_{k,j}) - \phi_k(x_{k,j}) - \mu_k s_{k,j}^T X_{k,j}^{-1}e - \frac{1}{2}s_{k,j}^T (B_{k,j} + \Theta_{k,j})s_{k,j}. \end{aligned}$$

Assume, as in Proposition 26, that $\|B_{k,j}\| \leq \tilde{M}_k$. Since $\Theta_{k,j}$ is bounded by definition, we

use (6.40) and (6.41) to conclude that $f(x_{k,j} + s_{k,j}) + \phi_k(x_{k,j} + s_{k,j}) - \varphi(s_{k,j}; x_{k,j}, B_{k,j}) = O(\|s_{k,j}\|^2)$, and, if $\psi(s; x_{k,j}) = h(x_{k,j} + s)$, (6.35a) holds. The proof of Proposition 27 is still valid when considering $B_{k,j} + \Theta_{k,j}$ instead of $B_{k,j} + X_{k,j}^{-1}Z_{k,j}$, so that (6.35b) also holds. As a consequence, Step Assumption 6.5.1 still holds with φ defined in (6.48).

Now, our goal is to find a sequence $\{w_k\} \rightarrow 0$ such that $w_k \in \nabla f(\bar{x}_k) - \bar{z}_k + \partial\psi(\bar{s}_{k,1}; \bar{x}_k)$. Under some additional assumptions on ψ , this will allow us to establish that Algorithm 6.5.1 generates iterates that satisfy asymptotically (6.9). We begin with preliminary lemmas.

Lemma 13. *Assume that $s_{k,j,1}$ is not on the boundary of $\Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})$ and that Model Assumption 6.5.1 holds. Then,*

$$-\nu_{k,j}^{-1}s_{k,j,1} + \mu_k X_{k,j}^{-1}e \in \nabla f(x_{k,j}) + \partial\psi(s_{k,j,1}; x_{k,j}). \quad (6.49)$$

Proof. The first step of the proximal gradient method $s_{k,j,1}$ satisfies (6.31). According to (6.18), its first-order optimality conditions are

$$0 \in s_{k,j,1} + \nu_{k,j} \nabla \varphi_{\text{cp}}(0; x_{k,j}) + \partial(\nu_{k,j} \psi(\cdot; x_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))) (s_{k,j,1}). \quad (6.50)$$

As $s_{k,j,1}$ is not on the boundary of $\Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})$, there exists $r > 0$ such that for all s in an open ball of center $s_{k,j,1}$ and radius r , $\chi(s \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = 0$. Thus, the definition of the subdifferential guarantees that

$$\begin{aligned} \partial(\nu_{k,j} \psi(\cdot; x_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))) (s_{k,j,1}) = \\ \nu_{k,j} \partial\psi(s_{k,j,1}; x_{k,j}) + \partial\chi(s_{k,j,1} \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})). \end{aligned}$$

We know that $\partial\chi(s_{k,j,1} \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = N_{\Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})}(s_{k,j,1})$ using Lemma 11, and $N_{\Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})}(s_{k,j,1}) = \{0\}$ because $s_{k,j,1}$ is not on the boundary of $\Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})$. Therefore, (6.50) simplifies to

$$0 \in \nu_{k,j}^{-1}s_{k,j,1} + \nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1}e + \partial\psi(s_{k,j,1}; x_{k,j})$$

using Model Assumption 6.5.1 for $\nabla \varphi_{\text{cp}}(0; x_{k,j})$. □

Lemma 14. *Let Parameter Assumption 6.5.1 be satisfied. Then, for all $k \in \mathbb{N}$,*

$$\|\bar{s}_{k,1}\| \leq \sqrt{2\bar{\nu}_k} \epsilon_{d,k}.$$

Proof. Since (6.33) holds, (6.45) leads to

$$\epsilon_{d,k} \geq \bar{\nu}_k^{-1/2} \xi_{\text{cp}}(\bar{\Delta}_k; \bar{x}_k, \bar{\nu}_k)^{1/2} \geq \frac{1}{\sqrt{2}} \bar{\nu}_k^{-1} \|\bar{s}_{k,1}\|,$$

which completes the proof. $\square$

The following assumption will be useful to establish that $\bar{s}_{k,1}$ converges to zero sufficiently fast to guarantee the convergence of the outer iterations.

Parameter Assumption 6.5.2. *The sequences $\{\epsilon_{d,k}\}$ used in Algorithm 6.5.1 and $\{\kappa_{\text{mdb},k}\}$ from Proposition 25 satisfy*

$$\kappa_{\text{mdb},k}^{-1} \epsilon_{d,k} \rightarrow 0. \quad (6.51)$$

To justify that Parameter Assumption 6.5.2 is reasonable, assume for simplicity that i is an index such that $(\bar{x}_k)_i = \kappa_{\text{mdb},k}$. We have

$$-e_i^T \bar{X}_k \bar{z}_k + \mu_k \leq \|\bar{X}_k \bar{z}_k - \mu_k e\| \leq \epsilon_{p,k},$$

so that

$$(\bar{x}_k)_i (\bar{z}_k)_i = e_i^T \bar{X}_k \bar{z}_k \geq \mu_k - \epsilon_{p,k},$$

and, if $\epsilon_{p,k} < \mu_k$,

$$\kappa_{\text{mdb},k}^{-1} = \frac{1}{(\bar{x}_k)_i} \leq \frac{(\bar{z}_k)_i}{\mu_k - \epsilon_{p,k}}.$$

We multiply the above inequality by $\epsilon_{p,k}$ to obtain

$$\kappa_{\text{mdb},k}^{-1} \epsilon_{p,k} \leq \frac{(\bar{z}_k)_i}{\mu_k \epsilon_{p,k}^{-1} - 1}.$$

If $\frac{\epsilon_{p,k}}{\mu_k} \rightarrow 0$, e.g., $\epsilon_{p,k} = \mu_k^{1+\gamma_k}$ with $0 < \gamma_k < 1$, $\kappa_{\text{mdb},k}^{-1} \epsilon_{p,k} \rightarrow 0$. Then, the choice $\epsilon_{d,k} = O(\epsilon_{p,k})$ guarantees that Parameter Assumption 6.5.2 is satisfied.

Lemma 15. *Let Model Assumption 6.5.1, Parameter Assumption 6.5.1 and Parameter Assumption 6.5.2 be satisfied, and for all j , $(f+h)(x_{k,j}) \geq (f+h)_{\text{low},k}$. Then, there exists $N \in \mathcal{N}_\infty$ such that for all $k \in N$, $\bar{s}_{k,1}$ is not on the boundary of $\bar{\Delta}_k \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(\bar{x}_k)$.*

Proof. For all $s \in \mathbb{R}^n$, $\|s\|_\infty \leq \|s\|$, thus Lemma 14 leads to $\|\bar{s}_{k,1}\|_\infty \leq \|\bar{s}_{k,1}\| \leq \sqrt{2} \bar{\nu}_k \epsilon_{d,k}$. As a consequence, if $\sqrt{2} \epsilon_{d,k} \bar{\nu}_k < \Delta_{\text{min},k}$, then $\bar{s}_{k,1}$ is not on the boundary of $\bar{\Delta}_k \mathbb{B}$. This is certainly true for k sufficiently large, because $\bar{\Delta} > 0$ in Parameter Assumption 6.5.1 and $\epsilon_{d,k} \rightarrow 0$.

Now, we show that $\bar{s}_{k,1}$ is not on the boundary of $\mathbb{R}_{\delta_k}^n(\bar{x}_k)$ if k is large enough. First, we point out that $\bar{\nu}_k \not\rightarrow \infty$, because

$$\nu_{k,j} = \frac{1}{\|B_{k,j}\| + \|\Theta_{k,j}\| + \alpha^{-1}\Delta_{k,j}^{-1}} \leq \alpha\Delta_{k,j}.$$

Then, we have

$$\begin{aligned} 0 \leq \frac{|\min_i(\bar{s}_{k,1})_i|}{\min_i(\bar{x}_k)_i} &\leq \frac{\|\bar{s}_{k,1}\|_\infty}{\min_i(\bar{x}_k)_i} \leq \frac{\|\bar{s}_{k,1}\|}{\min_i(\bar{x}_k)_i} \\ &\leq \frac{\sqrt{2}\bar{\nu}_k\epsilon_{d,k}}{\min_i(\bar{x}_k)_i} && \text{using Lemma 14} \\ &\leq \frac{\sqrt{2}\bar{\nu}_k\epsilon_{d,k}}{\kappa_{\text{mdb},k}} && \text{using Proposition 25,} \end{aligned}$$

and Parameter Assumption 6.5.2 indicates that $\sqrt{2}\kappa_{\text{mdb},k}^{-1}\bar{\nu}_k\epsilon_{d,k} \rightarrow 0$. As $\delta_k \rightarrow \bar{\delta} < 1$, the inequality

$$\frac{\min_i(\bar{s}_{k,1})_i}{\min_i(\bar{x}_k)_i} + 1 > \delta_k \quad (6.52)$$

is satisfied if k is large enough, and

$$\begin{aligned} \min_i(\bar{s}_{k,1} + \bar{x}_k)_i &\geq \min_i(\bar{s}_{k,1})_i + \min_i(\bar{x}_k) && \text{by properties of the min} \\ &> \delta_k \min_i(\bar{x}_k) && \text{with (6.52).} \end{aligned} \quad (6.53)$$

Therefore, $\bar{s}_{k,1}$ is not on the boundary of $\mathbb{R}_{\delta_k}(\bar{x}_k)$ if k is large enough. We conclude that there exists $N \in \mathcal{N}_\infty$ such that for all $k \in N$, $\bar{s}_{k,1}$ is not on the boundary of $\bar{\Delta}_k \mathbb{B} \cap \mathbb{R}_{\delta_k}(\bar{x}_k)$. $\square$

Theorem 5. *Let Model Assumption 6.5.1, Parameter Assumption 6.5.1 and Parameter Assumption 6.5.2 be satisfied, and for all j , $(f+h)(x_{k,j}) \geq (f+h)_{\text{low},k}$. In addition, suppose that $\epsilon_{p,k} = O(\epsilon_{d,k})$. We define*

$$w_k := -\bar{\nu}_k^{-1}\bar{s}_{k,1} + \mu_k\bar{X}_k^{-1}e - \bar{z}_k. \quad (6.54)$$

Then, there exists a subsequence $N \in \mathcal{N}_\infty$ such that for all $k \in N$,

$$w_k \in \nabla f(\bar{x}_k) - \bar{z}_k + \partial\psi(\bar{s}_{k,1}; \bar{x}_k), \quad (6.55)$$

and

$$\|w_k\| \leq \sqrt{2}\epsilon_{d,k} + \kappa_{\text{mdb},k}^{-1}\epsilon_{p,k} \rightarrow 0. \quad (6.56)$$

Proof. Lemma 15 indicates that there is a subsequence $N \in \mathcal{N}_\infty$ such that, for $k \in N$, $\bar{s}_{k,1}$ is not on the boundary of $\bar{\Delta}_k \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(\bar{x}_k)$. Thus, Lemma 13 holds, and

$$-\bar{\nu}_k^{-1}\bar{s}_{k,1} + \mu_k \bar{X}_k^{-1}e - \bar{z}_k \in \nabla f(\bar{x}_k) - \bar{z}_k + \partial\psi(\bar{s}_{k,1}; \bar{x}_k). \quad (6.57)$$

With w_k defined in (6.54), we have (6.55). Now, for all $i \in \{1, \dots, n\}$, we use Proposition 25 to establish that

$$(\mu_k \bar{X}_k^{-1}e - \bar{z}_k)_i = (\mu_k - (\bar{x}_k)_i(\bar{z}_k)_i)/(\bar{x}_k)_i \leq (\mu_k - (\bar{x}_k)_i(\bar{z}_k)_i)/\kappa_{\text{mdb},k},$$

and, by summing the square of the above inequality for all $i \in \{1, \dots, n\}$,

$$\begin{aligned} \|\mu_k \bar{X}_k^{-1}e - \bar{z}_k\|^2 &= \sum_{i=1}^n (\mu_k \bar{X}_k^{-1}e - \bar{z}_k)_i^2 \\ &\leq \sum_{i=1}^n (\mu_k - (\bar{x}_k)_i(\bar{z}_k)_i)^2 / \kappa_{\text{mdb},k}^2 \\ &= \|\mu_k e - \bar{X}_k \bar{z}_k\|^2 / \kappa_{\text{mdb},k}^2 \\ &\leq \kappa_{\text{mdb},k}^{-2} \epsilon_{p,k}^2 \text{ because (6.34) holds,} \end{aligned} \quad (6.58)$$

so that $\|\mu_k \bar{X}_k^{-1}e - \bar{z}_k\| \leq \kappa_{\text{mdb},k}^{-1} \epsilon_{p,k}$. As $\epsilon_{p,k} = O(\epsilon_{d,k})$ and Parameter Assumption 6.5.2 holds, we deduce that $\|\mu_k \bar{X}_k^{-1}e - \bar{z}_k\| \rightarrow 0$.

Finally,

$$\begin{aligned} \|w_k\| &= \|-\bar{\nu}_k^{-1}\bar{s}_{k,1} + \mu_k \bar{X}_k^{-1}e - \bar{z}_k\| \\ &\leq \bar{\nu}_k^{-1}\|\bar{s}_{k,1}\| + \|\mu_k \bar{X}_k^{-1}e - \bar{z}_k\| \\ &\leq \sqrt{2}\epsilon_{d,k} + \kappa_{\text{mdb},k}^{-1}\epsilon_{p,k} \xrightarrow{k \rightarrow +\infty} 0, \end{aligned}$$

where we used Lemma 14 and (6.58) in the last inequality. $\square$

Now, we present two assumptions on ψ . The first will not be necessary for the remaining results of this subsection, except as one of the justifications for the second assumption. However, it will be used in Section 6.5.4.

Model Assumption 6.5.3. $x \mapsto \psi(\cdot; x)$ is epi-continuous on $\mathbb{R}_+^n$.

Model Assumption 6.5.3 holds if ψ is continuous on $\mathbb{R}_+^n \times \mathbb{R}_+^n$, but this condition is only sufficient, not necessary [73, Exercise 7.40]. Let us consider the case where $\psi(\cdot; x) = s \mapsto h(x + s)$. Because h is lsc, its epigraph is closed, thus the sequence of functions $\{h, h, \dots\}$

satisfies $\{h, h, \dots\} \xrightarrow{e} h$. Let $\bar{x} \in \mathbb{R}_+^n$ and $\{x_k\} \rightarrow \bar{x}$. [73, Exercise 7.8d] indicates that for $h_k = s \mapsto h(x_k + s)$ and $\bar{h} = s \mapsto h(\bar{x} + s)$, $h_k \xrightarrow{e} \bar{h}$. Since the latter is true for all $\bar{x} \in \mathbb{R}_+^n$, we conclude that Model Assumption 6.5.3 is satisfied.

Model Assumption 6.5.4. *For any sequences $\{s_k\} \rightarrow 0$ and $\{x_k\} \rightarrow \bar{x} \geq 0$ such that $x_k + s_k > 0$ for all $k \in \mathbb{N}$,*

$$\limsup_{k \rightarrow \infty} \partial\psi(s_k; x_k) \subset \partial\psi(0; \bar{x}) = \partial h(\bar{x}). \quad (6.59)$$

We present some cases for which Model Assumption 6.5.4 holds.

- When $\text{g-lim}_{k \rightarrow \infty} \partial\psi(\cdot; x_k) = \partial\psi(\cdot; \bar{x})$. Attouch's theorem [5] (also written in [73, Theorem 12.35]) indicates that this condition is satisfied when $\psi(\cdot; x_k)$ and $\psi(\cdot; \bar{x})$ are proper, lsc, convex functions with $\psi(\cdot; x_k) \xrightarrow{e} \psi(\cdot; \bar{x})$ (i.e., Model Assumption 6.5.3 holds). An extension to non-convex functions under some more sophisticated assumptions is established by Poliquin [72].
- When $\psi(s; x) = h(x + s)$ and $h(x_k + s_k) \rightarrow h(\bar{x})$ (e.g., $h = \|\cdot\|_1$), using Proposition 20 applied to $\{x_k + s_k\} \rightarrow \bar{x}$.
- When $\psi(s; x) = h(x + s)$ but h is not continuous, we may still be able to show that Model Assumption 6.5.4 holds. For example, with $h = \|\cdot\|_0$, [53, Theorem 1] shows that $\partial h(x) = \{v \mid v_i = 0 \text{ if } x_i \neq 0\}$. Thus, $\partial\psi(s_k; x_k) = \partial(\|\cdot\|_0)(x_k + s_k) = \{0\} \subset \partial h(\bar{x})$.

Corollary 2. *Under the assumptions of Theorem 5 and Model Assumption 6.5.4, let $\bar{x}$ and $\bar{z}$ be limit points of $\{\bar{x}_k\}$ and $\{\bar{z}_k\}$, respectively. Then,*

$$0 \in \nabla f(\bar{x}) - \bar{z} + \partial h(\bar{x}) \quad \text{and} \quad \bar{X} \bar{Z} e = 0. \quad (6.60)$$

In this case, when the CQ is satisfied at $\bar{x}$, $\bar{x}$ is first-order stationary for (6.1).

Proof. In Theorem 5, N can be chosen such that $\bar{x}_k \xrightarrow[k \in N]{} \bar{x}$ and $\bar{z}_k \xrightarrow[k \in N]{} \bar{z}$. We apply Model Assumption 6.5.4 to (6.55) and (6.56) to deduce

$$-\nabla f(\bar{x}) + \bar{z} \in \partial\psi(0; \bar{x}) = \partial h(\bar{x}), \quad (6.61)$$

which indicates that $0 \in P^{\mathcal{L}}(\bar{\Delta}; \bar{x}, \bar{z}, \bar{\nu})$. The condition (6.34) implies that $\bar{X} \bar{Z} e = 0$. When the CQ is satisfied, we can use Lemma 9 to conclude that $\bar{x}$ is first-order stationary for (6.1). $\square$

In Theorem 5, w_k evokes of the concept of (ϵ_p, ϵ_d) -KKT optimality for interior-point methods introduced in [37, Definition 2.1]. We slightly modify this concept in the following definition.

Definition 2. Let $\epsilon_p, \epsilon_d \geq 0$, $x \geq 0$, and $\Delta, \nu > 0$. x is said to be (ϵ_p, ϵ_d) -KKT optimal if there exist $z \geq 0$ and $v \in \partial h(x)$ such that

$$\|\nabla f(x) - z + v\| \leq \epsilon_d, \quad (6.62)$$

and, for all $i \in \{1, \dots, n\}$,

$$x_i z_i \leq \epsilon_p. \quad (6.63)$$

The main modification to the original formulation in [37, Definition 2.1] is that we require $x_i z_i \leq \epsilon_p$ instead of $\min(x_i, z_i) \leq \epsilon_p$ for all i , but this is linked to our different choices of stopping condition for the complementary slackness. The first part of the definition (6.62) is similar to the ϵ -stationarity [36, Definition 4.5] for more general problems.

Model Assumption 6.5.1 does not necessarily guarantee that $\partial\psi(s_{k,j,1}; x_{k,j}) = \partial h(x_{k,j} + s_{k,j,1})$. Thus, for $k \in N$ where N is a subsequence introduced in Theorem 5, we cannot use Theorem 5 to measure the (ϵ_p, ϵ_d) -KKT optimality of $\bar{x}_k$. Let

$$\epsilon_{h,k} = \text{dist}(w_k - \nabla f(\bar{x}_k) + \bar{z}_k, \partial h(\bar{x}_k + \bar{s}_{k,1})). \quad (6.64)$$

As Theorem 5 indicates that $w_k - f(\bar{x}_k) + \bar{z}_k \in \partial\psi(\bar{s}_{k,1}; \bar{x}_k)$, we can obtain a measure of (ϵ_p, ϵ_d) -KKT optimality which depends on $\epsilon_{h,k}$. When all the elements of $\partial\psi(\bar{s}_{k,1}; \bar{x}_k)$ are close to an element of $\partial h(\bar{x}_k + \bar{s}_{k,1})$, we expect $\epsilon_{h,k}$ to be small. In particular, if $\partial\psi(\bar{s}_{k,1}; \bar{x}_k) \subseteq \partial h(\bar{x}_k + \bar{s}_{k,1})$, $\epsilon_{h,k} = 0$.

Theorem 6. Let the assumptions of Theorem 5 be satisfied, and $\epsilon_{h,k}$ be defined in (6.64). Then, there exists $N \in \mathcal{N}_\infty$ such that for all $k \in N$, $\bar{x}_k + \bar{s}_{k,1}$ is $(\bar{\epsilon}_{p,k}, \bar{\epsilon}_{d,k})$ -KKT optimal with constants

$$\begin{aligned} \bar{\epsilon}_{p,k} &= \epsilon_{p,k} + \sqrt{n}\mu_k + \sqrt{2}\bar{\nu}_k\epsilon_{d,k}\|\bar{z}_k\| \\ \bar{\epsilon}_{d,k} &= \epsilon_{h,k} + \sqrt{2}\epsilon_{d,k}(1 + \bar{\nu}_k L_f) + \kappa_{\text{mdb},k}^{-1}\epsilon_{p,k}. \end{aligned}$$

Proof. Theorem 5 guarantees that, for all k in a subsequence $N \in \mathcal{N}_\infty$, (6.55) holds, i.e.

$$w_k - \nabla f(\bar{x}_k) + \bar{z}_k \in \partial\psi(\bar{s}_{k,1}; \bar{x}_k),$$

where w_k is defined in (6.54). As $\partial h(\bar{x}_k + \bar{s}_{k,1})$ is closed, we can choose $v_k \in \partial h(\bar{x}_k + \bar{s}_{k,1})$ such that, for $y_k := v_k - w_k + \nabla f(\bar{x}_k) - \bar{z}_k$, we have $\|y_k\| = \epsilon_{h,k}$. Then,

$$(v_k - w_k + \nabla f(\bar{x}_k) - \bar{z}_k) + w_k \in \nabla f(\bar{x}_k) - \bar{z}_k + \partial h(\bar{x}_k + \bar{s}_{k,1}),$$

which we may rewrite as

$$y_k + w_k + \nabla f(\bar{x}_k + \bar{s}_{k,1}) - \nabla f(\bar{x}_k) \in \nabla f(\bar{x}_k + \bar{s}_{k,1}) - \bar{z}_k + \partial h(\bar{x}_k + \bar{s}_{k,1}),$$

and

$$\|y_k + w_k + \nabla f(\bar{x}_k + \bar{s}_{k,1}) - \nabla f(\bar{x}_k)\| \leq \epsilon_{h,k} + \|w_k\| + L_f \|\bar{s}_{k,1}\|.$$

Lemma 14 implies that $L_f \|\bar{s}_{k,1}\| \leq \sqrt{2} L_f \bar{\nu}_k \epsilon_{d,k}$. We combine the latter inequality with (6.56) in Theorem 5 to obtain

$$\|y_k + w_k + \nabla f(\bar{x}_k + \bar{s}_{k,1}) - \nabla f(\bar{x}_k)\| \leq \epsilon_{h,k} + \sqrt{2} \epsilon_{d,k} (1 + \bar{\nu}_k L_f) + \kappa_{\text{mdb},k}^{-1} \epsilon_{p,k}. \quad (6.65)$$

Now, for all $i \in \{1, \dots, n\}$,

$$\begin{aligned} (\bar{x}_k + \bar{s}_{k,1})_i (\bar{z}_k)_i &\leq \|(\bar{X}_k + \bar{S}_{k,1}) \bar{z}_k\| \\ &\leq \|\bar{X}_k \bar{z}_k - \mu_k e\| + \|\mu_k e\| + \|\bar{S}_{k,1} \bar{z}_k\| \\ &\leq \epsilon_{p,k} + \sqrt{n} \mu_k + \|\bar{s}_{k,1}\| \|\bar{z}_k\| \\ &\leq \epsilon_{p,k} + \sqrt{n} \mu_k + \sqrt{2} \bar{\nu}_k \epsilon_{d,k} \|\bar{z}_k\|. \end{aligned} \quad (6.66)$$

We use (6.65), (6.66) and Definition 2 to conclude. $\square$

If $\bar{z}_k$ is bounded, $\bar{\epsilon}_{p,k} \xrightarrow[N]{} 0$ in Theorem 6. If $\epsilon_{h,k} \rightarrow 0$, we also have $\bar{\epsilon}_{d,k} \xrightarrow[N]{} 0$.

6.5.4 Convergence with a new criticality measure

Now, instead of using $\nu_{k,j}^{-1/2} \xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2}$ (involving $\varphi_{\text{cp}}(\cdot; x_{k,j})$) for the criticality measure of Algorithm 6.5.2, we would like to use a measure based upon $\varphi^{\mathcal{L}}(\cdot; x_{k,j}, z_{k,j})$ defined in (6.10a). The reason behind this choice is inspired from the criticality measure $\|\nabla f(x_{k,j}) - z_{k,j}\|$ used in primal-dual trust region algorithms in the smooth case, instead of $\|\nabla f(x_{k,j}) - \mu_k X_{k,j}^{-1} e\|$ used in primal algorithms, see for example [34, Algorithm 13.6.2]. We may expect that this choice results in fewer iterations of Algorithm 6.5.2 when $x_{k,j}$ and $z_{k,j}$ are close to a solution of (6.1), because (we express this idea with smooth notations for now), if j_k is the index for which the stopping criteria of Algorithm 6.5.2 are met, $\|\nabla f(\bar{x}_k) - \bar{z}_k\| = \|\nabla f(x_{k+1,0}) - z_{k+1,0}\|$, whereas $\|\nabla f(\bar{x}_k) - \mu_k \bar{X}_k^{-1} e\| \neq \|\nabla f(x_{k+1,0}) - \mu_{k+1} X_{k+1,0}^{-1} e\|$. However, to change the stopping criterion, we need the following convexity assumption.

Model Assumption 6.5.5. *For a sequence $\{x_{k,j}\}_j$ generated by Algorithm 6.5.2 at iteration k , $\psi(\cdot; x_{k,j})$ is convex for all j .*

If $\psi(s; x) = h(x + s)$ and h is convex, Model Assumption 6.5.5 holds.

In this section, we define

$$s_{k,j}^{\mathcal{L}} \in \underset{s}{\operatorname{argmin}} m^{\mathcal{L}}(s; x_{k,j}, z_{k,j}, \nu_{k,j}) + \chi(s \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})), \quad (6.67)$$

where $m^{\mathcal{L}}(s; x_{k,j}, z_{k,j}, \nu_{k,j})$ is defined in (6.11), and

$$\xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,j}; x_{k,j}, z_{k,j}, \nu_{k,j}) = (f + \phi_k + h)(x_{k,j}) - (\varphi^{\mathcal{L}}(s_{k,j}^{\mathcal{L}}; x_{k,j}, z_{k,j}) - \psi(s_{k,j}^{\mathcal{L}}; x_{k,j})), \quad (6.68)$$

where $\varphi^{\mathcal{L}}(\cdot; x_{k,j}, z_{k,j})$ is defined in (6.10a). We point out that $\xi_{\delta_k}^{\mathcal{L}}$ and $\xi^{\mathcal{L}}$ defined in (6.13) are almost identical, the latter being computed by replacing $\chi(s \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))$ by $\chi(s \mid \Delta_{k,j}\mathbb{B}) + \chi(x_{k,j} + s \mid \mathbb{R}_+^n)$ in (6.67). Model Assumption 6.5.5 will be useful in Theorem 7, which is crucial for our analysis of Algorithm 6.5.3 because it establishes the convergence of the inner iterations with $\xi_{\delta_k}^{\mathcal{L}}$.

Algorithm 6.5.3 resembles Algorithm 6.5.1, except for the stopping criterion (6.33). Our ultimate goal in this subsection is to show that replacing (6.33) in Algorithm 6.5.1 by (6.69) in Algorithm 6.5.3 maintains similar convergence properties to those of Section 6.5.3, but, as illustrated in Section 6.6, performs better in practice.

Algorithm 6.5.3 Nonsmooth interior-point method (outer iteration) with stopping criteria based upon $\xi_{\delta_k}^{\mathcal{L}}$ in (6.68).

- 1: Choose $\epsilon > 0$, sequences $\{\mu_k\} \searrow 0$, $\{\epsilon_{d,k}\} \searrow 0$, $\{\epsilon_{p,k}\} \searrow 0$, and $\{\delta_k\} \rightarrow \bar{\delta} \in [0, 1)$ with $\delta_k \in (0, 1)$ for all k .
- 2: Choose $x_{0,0} \in \mathbb{R}_{++}^n$ where h is finite.
- 3: **for** $k = 0, 1, \dots$ **do**
- 4: Compute an approximate solution $x_k := x_{k,j}$ to (6.22) and $z_k := z_{k,j}$ in the sense that

$$\nu_{k,j}^{-1/2} \xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j})^{1/2} \leq \epsilon_{d,k}, \quad (6.69)$$

and (6.34) holds, that we recall in the following inequality for conveniency

$$\|\mu_k e - X_{k,j} z_{k,j}\| \leq \epsilon_{p,k}.$$

- 5: Set $x_{k+1,0} := x_k$.
-

The following result shows that the inner iteration terminates finitely.

Theorem 7. *Under the assumptions of Proposition 30, Model Assumption 6.5.3 and Model*

Assumption 6.5.5, if $\{x_{k,j}\}_j$ possesses a limit point x_k^* , then

$$\liminf_{j \rightarrow +\infty} \xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,j}; x_{k,j}, z_{k,j}, \nu_{k,j}) = 0, \quad (6.70)$$

and

$$\liminf_{j \rightarrow +\infty} \|s_{k,j}^{\mathcal{L}}\| = 0, \quad (6.71)$$

where $s_{k,j}^{\mathcal{L}}$ is defined in (6.67).

Proof. As $\Delta_{k,j} \in [\Delta_{\min,k}, \Delta_{\max}]$ and $\nu_{k,j} \in [\nu_{\min,k}, 1]$, there exists an infinite subsequence N such that $\Delta_{k,j} \xrightarrow{j \in N} \Delta_k^*$, $\nu_{k,j} \xrightarrow{j \in N} \nu_k^*$, $x_{k,j} \xrightarrow{j \in N} x_k^*$, and with Proposition 30 $z_{k,j} \xrightarrow{j \in N} z_k^*$ with $X_k^* Z_k^* e = \mu_k e$. By continuity of the min, $\mathbb{R}_{\delta_k}^n(x_{k,j}) \xrightarrow{j \in N} \mathbb{R}_{\delta_k}^n(x_k^*)$. The sets $\Delta_{k,j}\mathbb{B}$ and $\mathbb{R}_{\delta_k}^n(x_{k,j})$ are convex (using Lemma 11 for the latter). Since $\Delta_k^*\mathbb{B}$ and $\mathbb{R}_{\delta_k}^n(x_k^*)$ are convex and cannot be separated, we use [73, Theorem 4.33] to conclude that

$$\Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}) \xrightarrow{j \in N} \Delta_k^*\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_k^*).$$

With [73, Theorem 7.4f], we deduce

$$\text{e-lim}_{j \in N} \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = \chi(\cdot \mid \Delta_k^*\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_k^*)).$$

Thanks to Proposition 30 and the smoothness of $\varphi_{\text{cp}}(\cdot; x)$ and $\varphi^{\mathcal{L}}(\cdot; x, z)$, we also have

$$\text{e-lim}_{j \in N} \varphi_{\text{cp}}(\cdot; x_{k,j}) = \text{e-lim}_{j \in N} \varphi^{\mathcal{L}}(\cdot; x_{k,j}, z_{k,j}) = \varphi^{\mathcal{L}}(\cdot; x_k^*, z_k^*). \quad (6.72)$$

The functions $\varphi_{\text{cp}}(\cdot; x_{k,j})$, $\varphi^{\mathcal{L}}(\cdot; x_{k,j}, z_{k,j})$ and $\varphi^{\mathcal{L}}(\cdot; x_k^*, z_k^*)$ are all convex because they are linear. Model Assumption 6.5.3 implies that $\text{e-lim}_{j \in N} \psi(\cdot; x_{k,j}) = \psi(\cdot; x_k^*)$. Model Assumption 6.5.5 and [73, Theorem 7.46] lead to

$$\text{e-lim}_{j \in N} \psi(\cdot; x_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = \psi(\cdot; x_k^*) + \chi(\cdot \mid \Delta_k^*\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_k^*)), \quad (6.73)$$

and the above functions are all convex. We deduce from (6.72), (6.73) and [73, Theorem 7.46] that

$$\text{e-lim}_{j \in N} m_{\text{cp}}(\cdot; x_{k,j}, \nu_{k,j}) = \text{e-lim}_{j \in N} m^{\mathcal{L}}(\cdot; x_{k,j}, z_{k,j}, \nu_{k,j}) = m^{\mathcal{L}}(\cdot; x_k^*, z_k^*, \nu_k^*),$$

where $m^{\mathcal{L}}$ is defined in (6.11). The sequences $m_{\text{cp}}(\cdot; x_{k,j}, \nu_{k,j}) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))$ and $m^{\mathcal{L}}(\cdot; x_k^*, z_k^*, \nu_k^*) + \chi(\cdot \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))$ are level-bounded because of the indicators. As

in (6.31), we have

$$\begin{aligned} \text{prox}_{\nu_k^* \psi(\cdot; x_k^*) + \chi(\cdot | \Delta_k^* \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_k^*))}(-\nu_k^* \nabla \varphi_{\text{cp}}(0; x_k^*, \nu_k^*)) = \\ \arg \min_s m^{\mathcal{L}}(s; x_k^*, z_k^*, \nu_k^*) + \chi(s | \Delta_k^* \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_k^*)), \end{aligned}$$

and the above problem is single valued because of [73, Theorem 2.26a]. Let s^* denote its only solution. Theorem 3, and specifically (6.5), implies that the sequences

$$s_{k,j,1} \in \arg \min_s m_{\text{cp}}(s; x_{k,j}, \nu_{k,j}) + \chi(s | \Delta_{k,j} \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})),$$

and

$$s_{k,j}^{\mathcal{L}} \in \arg \min_s m^{\mathcal{L}}(s; x_{k,j}, z_{k,j}, \nu_{k,j}) + \chi(s | \Delta_{k,j} \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j}))$$

have the same limit s^* . We have shown in Lemma 12 that $s_{k,j,1} \xrightarrow{j \rightarrow +\infty} 0$. Thus, $s^* = 0$. Finally, we have

$$\xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j}) = (f + h + \phi_k)(x_{k,j}) - m_{\text{cp}}(s_{k,j,1}; x_{k,j}, \nu_{k,j}) + \frac{1}{2} \nu_{k,j}^{-1} \|s_{k,j,1}\|^2.$$

As $\xi_{\text{cp}}(\Delta_{k,j}; x_{k,j}, \nu_{k,j}) \xrightarrow{j \rightarrow +\infty} 0$ by Proposition 29, and $\|s_{k,j,1}\| \xrightarrow{j \rightarrow +\infty} 0$, we deduce that

$$m_{\text{cp}}(s_{k,j,1}; x_{k,j}, \nu_{k,j}) \xrightarrow{j \in N} (f + h + \phi_k)(x_k^*).$$

Using (6.4) in Theorem 3, we have

$$\lim_{j \in N} m_{\text{cp}}(s_{k,j,1}; x_{k,j}, \nu_{k,j}) = \lim_{j \in N} m^{\mathcal{L}}(s_{k,j}^{\mathcal{L}}; x_{k,j}, z_{k,j}, \nu_{k,j}) = (f + h + \phi_k)(x_k^*). \quad (6.74)$$

The expression of $\xi_{\delta_k}^{\mathcal{L}}$ in (6.68) can also be written as

$$\xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,j}; x_{k,j}, z_{k,j}, \nu_{k,j}) = (f + h + \phi_k)(x_{k,j}) - m^{\mathcal{L}}(s_{k,j}^{\mathcal{L}}; x_{k,j}, z_{k,j}, \nu_{k,j}) + \frac{1}{2} \nu_{k,j}^{-1} \|s_{k,j}^{\mathcal{L}}\|^2.$$

By injecting the limit of $s_{k,j}^{\mathcal{L}}$ and (6.74) in the above equation, we obtain (6.70). $\square$

From this point on, j_k denotes the number of iterations performed by Algorithm 6.5.2 at iteration k with the inner stopping criteria from Algorithm 6.5.3, and we use again the notation $\bar{x}_k = x_{k,j_k}$, $\bar{z}_k = z_{k,j_k}$, $\bar{s}_{k,1} = s_{k,j_k,1}$, $\bar{\Delta}_k = \Delta_{k,j_k}$, $\bar{\nu}_k = \nu_{k,j_k}$, with the addition of $\bar{s}_k^{\mathcal{L}} := s_{k,j_k}^{\mathcal{L}}$. The following three lemmas are analogous to Lemma 13, Lemma 14 and Lemma 15.

Lemma 16. *Assume that $s_{k,j}^{\mathcal{L}}$ is not on the boundary of $\Delta_{k,j} \mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})$ and that Model*

Assumption 6.5.1 holds. Then,

$$-\nu_{k,j}^{-1} s_{k,j}^{\mathcal{L}} \in \nabla f(x_{k,j}) - z_{k,j} + \partial\psi(s_{k,j}^{\mathcal{L}}; x_{k,j}). \quad (6.75)$$

Proof. The first-order stationarity condition of (6.67) is

$$-\nu_{k,j}^{-1} s_{k,j}^{\mathcal{L}} \in \nabla f(x_{k,j}) - z_{k,j} + \partial\psi(s_{k,j}^{\mathcal{L}}; \bar{x}_k) + \partial\chi(s_{k,j}^{\mathcal{L}} \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})).$$

The same analysis as in the proof of Lemma 13 establishes that

$$\partial\chi(s_{k,j}^{\mathcal{L}} \mid \Delta_{k,j}\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(x_{k,j})) = \{0\},$$

so that (6.75) holds. $\square$

Lemma 17. *Let Parameter Assumption 6.5.1 be satisfied. Then, for all $k \in \mathbb{N}$,*

$$\|\bar{s}_k^{\mathcal{L}}\| \leq \sqrt{2}\bar{\nu}_k \epsilon_{d,k}.$$

Proof. The bound

$$\xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,j}; x_{k,j}, z_{k,j}, \nu_{k,j}) \geq \frac{1}{2} \nu_{k,j}^{-1} \|s_{k,j}^{\mathcal{L}}\|^2 \quad (6.76)$$

holds because

$$\begin{aligned} m^{\mathcal{L}}(0; x_{k,j}, z_{k,j}, \nu_{k,j}) &= (f + \phi_k + h)(x_{k,j}) \\ &\geq m^{\mathcal{L}}(s_{k,j}^{\mathcal{L}}; x_{k,j}, z_{k,j}, \nu_{k,j}) \\ &= (f + \phi_k + h)(x_{k,j}) + (\nabla f(x_{k,j}) + z_{k,j})^T s_{k,j}^{\mathcal{L}} + \frac{1}{2} \nu_{k,j}^{-1} \|s_{k,j}^{\mathcal{L}}\|^2. \end{aligned}$$

The stopping criterion (6.69) and (6.76) lead to

$$\frac{1}{\sqrt{2}} \bar{\nu}_k^{-1} \|\bar{s}_k^{\mathcal{L}}\| \leq \bar{\nu}_k^{-1/2} \xi_{\delta_k}^{\mathcal{L}}(\bar{\Delta}_k; \bar{x}_k, \bar{z}_k, \bar{\nu}_k)^{1/2} \leq \epsilon_{d,k},$$

which completes the proof. $\square$

Lemma 18. *Let Model Assumption 6.5.1, Parameter Assumption 6.5.1 and Parameter Assumption 6.5.2 be satisfied, and for all j , $(f + h)(x_{k,j}) \geq (f + h)_{\text{low},k}$. Then, there exists $N \in \mathcal{N}_{\infty}$ such that for all $k \in N$, $\bar{s}_k^{\mathcal{L}}$ is not on the boundary of $\bar{\Delta}_k\mathbb{B} \cap \mathbb{R}_{\delta_k}^n(\bar{x}_k)$.*

Proof. Since for any $s \in \mathbb{R}^n$, $\|s\|_{\infty} \leq \|s\|$, Lemma 17 leads to $\|\bar{s}_k^{\mathcal{L}}\|_{\infty} \leq \|\bar{s}_k^{\mathcal{L}}\| \leq \sqrt{2}\bar{\nu}_k \epsilon_{d,k}$. Thus, if $\sqrt{2}\epsilon_{d,k}\bar{\nu}_k < \Delta_{\min,k}$, $\bar{s}_k^{\mathcal{L}}$ is not on the boundary of $\bar{\Delta}_k\mathbb{B}$. As $\bar{\Delta} > 0$ in Parameter

Assumption 6.5.1 and $\epsilon_{d,k} \rightarrow 0$, this is true if k is sufficiently large. The rest of the proof is identical to that of Lemma 15. $\square$

Now, we can establish results similar to Theorem 5, Corollary 2, and Theorem 6 for Algorithm 6.5.3.

Theorem 8. *Let Model Assumption 6.5.1, Parameter Assumption 6.5.1 and Parameter Assumption 6.5.2 be satisfied, and for all j , $(f + h)(x_{k,j}) \geq (f + h)_{\text{low},k}$. Then, there exists a subsequence $N \in \mathcal{N}_\infty$ such that for all $k \in N$,*

$$-\bar{\nu}_k^{-1} \bar{s}_k^\mathcal{L} \in \nabla f(\bar{x}_k) - \bar{z}_k + \partial\psi(\bar{s}_k^\mathcal{L}; \bar{x}_k), \quad (6.77)$$

with

$$\nu_{k,j}^{-1} \|\bar{s}_k^\mathcal{L}\| \leq \sqrt{2}\epsilon_{d,k} \rightarrow 0. \quad (6.78)$$

Proof. Lemma 16 and Lemma 18 lead to (6.77). Lemma 17 shows that $\nu_{k,j}^{-1} \|s_{k,j}^\mathcal{L}\| \leq \sqrt{2}\epsilon_{d,k}$, and, as $\epsilon_{d,k} \rightarrow 0$, (6.78) is satisfied. $\square$

Corollary 3. *Under the assumptions of Theorem 8 and Model Assumption 6.5.4, let $\bar{x}$ and $\bar{z}$ be limit points of $\{\bar{x}_k\}$ and $\{\bar{z}_k\}$, respectively. Then,*

$$0 \in \nabla f(\bar{x}) - \bar{z} + \partial h(\bar{x}) \quad \text{and} \quad \bar{X} \bar{Z} e = 0. \quad (6.79)$$

In this case, when the CQ is satisfied, $\bar{x}$ is first-order stationary for (6.1).

Proof. In Theorem 8, N can be chosen such that $\bar{x}_k \xrightarrow[k \in N]{} \bar{x}$ and $\bar{z}_k \xrightarrow[k \in N]{} \bar{z}$. We apply Model Assumption 6.5.4 to (6.77) and (6.78) to obtain

$$-\nabla f(\bar{x}) + \bar{z} \in \partial\psi(0; \bar{x}) = \partial h(\bar{x}),$$

and we conclude as in the proof of Corollary 2. $\square$

Finally, we show a result similar to Theorem 6 for the (ϵ_p, ϵ_d) -KKT optimality. We use again a subsequence N as in theorem 8, $v_k \in \partial h(\bar{x}_k + \bar{s}_k^\mathcal{L})$, and we define

$$\epsilon_{h,k}^\mathcal{L} = \text{dist}(-\bar{\nu}_k^{-1} \bar{s}_k^\mathcal{L} - \nabla f(\bar{x}_k) + \bar{z}_k, \partial h(\bar{x}_k + \bar{s}_{k,1}^\mathcal{L})). \quad (6.80)$$

Theorem 9. *Let the assumptions of Theorem 8 be satisfied, and $\epsilon_{h,k}^\mathcal{L}$ be defined in (6.80). Then, there exists a subsequence $N \in \mathcal{N}_\infty$ such that for all $k \in N$, $\bar{x}_k + \bar{s}_k^\mathcal{L}$ is $(\bar{\epsilon}_{p,k}^\mathcal{L}, \bar{\epsilon}_{d,k}^\mathcal{L})$ -KKT*

optimal with constants

$$\begin{aligned}\bar{\epsilon}_{p,k}^{\mathcal{L}} &= \epsilon_{p,k} + \sqrt{n}\mu_k + \sqrt{2}\bar{\nu}_k\epsilon_{d,k}\|\bar{z}_k\| \\ \bar{\epsilon}_{d,k}^{\mathcal{L}} &= \epsilon_{h,k}^{\mathcal{L}} + \sqrt{2}\epsilon_{d,k}(1 + L_f\bar{\nu}_k).\end{aligned}$$

Proof. Theorem 8 guarantees that there exists an infinite subsequence N such that for all $k \in N$,

$$-\bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} - \nabla f(\bar{x}_k) + \bar{z}_k \in \partial\psi(\bar{s}_k^{\mathcal{L}}; \bar{x}_k),$$

Since $\partial h(\bar{x}_k + \bar{s}_k^{\mathcal{L}})$ is closed and nonempty, we can choose $v_k \in \partial h(\bar{x}_k + \bar{s}_k^{\mathcal{L}})$ such that, for $y_k^{\mathcal{L}} := -\bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} - \nabla f(\bar{x}_k) + \bar{z}_k$, we have $\|y_k^{\mathcal{L}}\| = \epsilon_{h,k}^{\mathcal{L}}$. Now,

$$(v_k - (-\bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} - \nabla f(\bar{x}_k) + \bar{z}_k)) - \bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} \in \nabla f(\bar{x}_k) - \bar{z}_k + \partial h(\bar{x}_k + \bar{s}_k^{\mathcal{L}}),$$

which can also be written as

$$y_k^{\mathcal{L}} - \bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} + \nabla f(\bar{x}_k + \bar{s}_k^{\mathcal{L}}) - \nabla f(\bar{x}_k) \in \nabla f(\bar{x}_k + \bar{s}_k^{\mathcal{L}}) - \bar{z}_k + \partial h(\bar{x}_k + \bar{s}_k^{\mathcal{L}}).$$

The triangle inequality combined and the Lipschitz constant L_f of ∇f leads to

$$\|y_k^{\mathcal{L}} - \bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} + \nabla f(\bar{x}_k + \bar{s}_k^{\mathcal{L}}) - \nabla f(\bar{x}_k)\| \leq \epsilon_{h,k}^{\mathcal{L}} + \bar{\nu}_k^{-1}\|\bar{s}_k^{\mathcal{L}}\| + L_f\|\bar{s}_k^{\mathcal{L}}\|.$$

Lemma 17 then implies that

$$\|y_k^{\mathcal{L}} - \bar{\nu}_k^{-1}\bar{s}_k^{\mathcal{L}} + \nabla f(\bar{x}_k + \bar{s}_k^{\mathcal{L}}) - \nabla f(\bar{x}_k)\| \leq \epsilon_{h,k}^{\mathcal{L}} + \sqrt{2}\epsilon_{d,k}(1 + L_f\bar{\nu}_k).$$

Finally, the inequalities of (6.66) still hold when replacing $\bar{s}_{k,1}$ by $\bar{s}_k^{\mathcal{L}}$: for all $i \in \{1, \dots, n\}$, $(\bar{x}_k)_i(\bar{z}_k)_i \leq \epsilon_{p,k} + \sqrt{n}\mu_k + \sqrt{2}\bar{\nu}_k\epsilon_{d,k}\|\bar{z}_k\|$. $\square$

When $\bar{x}_k \xrightarrow[k \in N]{} \bar{x} \in \mathbb{R}_+^n$ and $\bar{z}_k \xrightarrow[k \in N]{} \bar{z} \in \mathbb{R}_+^n$, $\{\bar{z}_k\}_N$ is bounded, thus $\bar{\epsilon}_{p,k}^{\mathcal{L}} \xrightarrow[N]{} 0$ in Theorem 9. If $\epsilon_{h,k}^{\mathcal{L}} \rightarrow 0$, we also have $\bar{\epsilon}_{d,k}^{\mathcal{L}} \xrightarrow[N]{} 0$.

6.6 Implementation and numerical experiments

All solvers tested are available from RegularizedOptimization.jl. We define

$$\epsilon_{d,k} := \epsilon_k + \epsilon_{r,i}\nu_{k,0}^{-1/2}\xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,0}; x_{k,0}, z_{k,0}, \nu_{k,0})^{1/2} \quad \text{and} \quad \epsilon_{p,k} := \epsilon_k,$$

where $\epsilon_{r,i} \geq 0$ is a predefined relative tolerance for the inner iterations. Algorithm 6.5.2 terminates when

$$\begin{aligned} \nu_{k,j}^{-1/2} \xi_{\delta_k}^{\mathcal{L}}(\Delta_{k,j}; x_{k,j}, z_{k,j}, \nu_{k,j_k})^{1/2} &< \epsilon_{d,k} \\ \|X_{k,j} z_{k,j} - \mu_k e\| &< \epsilon_{p,k}. \end{aligned}$$

We use the constant $\kappa_{\text{bar}} = 10^6$ for $\Theta_{k,j}$ in (6.48) and $\epsilon_{r,i} = 10^{-1}$. We set $\epsilon_0 = \mu_0 = 1$, $\mu_{k+1} = \mu_k/10$, $\epsilon_k = \mu_k^{1.01}$, and $\Delta_{k,0} = 1000\mu_k$, similarly as Conn *et al.* [33] did in the smooth case for their interior-point trust-region algorithm. In addition, we go to iteration $k+1$ and set $x_{k+1,0} = x_{k,j}$ if Algorithm 6.5.2 performs more than $j = 200$ iterations. Any iteration k where $x_{k,0}$ is not first-order stationary for (6.1) has $\epsilon_{d,k} > \epsilon_{p,k} = \epsilon_k$, thus the remarks below Parameter Assumption 6.5.2 are not sufficient to prove that this assumption is always satisfied, however, we observe satisfying performance with these parameters. Although it is possible to use $\epsilon_{d,k} = \epsilon_k$, it would require more inner iterations.

To declare convergence of Algorithm 6.5.3, we use the following criteria

$$\mu_k < \epsilon \tag{6.81a}$$

$$\|\bar{X}_k \bar{z}_k - \mu_k e\| < \epsilon \tag{6.81b}$$

$$\bar{\nu}_k^{-1/2} \xi_{\delta_k}^{\mathcal{L}}(\bar{\Delta}_k; \bar{x}_k, \bar{z}_k, \bar{\nu}_k)^{1/2} < \epsilon, \tag{6.81c}$$

where $\epsilon = \epsilon_a + \epsilon_r \nu_{1,1}^{-1/2} \xi_{\delta_1}^{\mathcal{L}}(\Delta_{1,1}; x_{1,1}, \nu_{1,1})^{1/2}$, for some $\epsilon_a \geq 0$. In our experiments, we chose $\epsilon_a = \epsilon_r = 10^{-4}$. We could not base our criteria upon $\xi^{\mathcal{L}}$ in (6.13) because we do not know the ν associated to this measure (which is different from the $\nu_{k,j}$ generated by Algorithm 6.5.2 associated to the barrier subproblem).

Once Algorithm 6.5.3 terminates, we use a crossover technique to set $x_i = 0$ or/and $z_i = 0$, to respect the complementarity condition. To do so, we check the final value of x_i (resp. z_i), and if it is smaller than $\sqrt{\mu}$ we set it to zero. If both x_i and z_i are smaller than $\mu^{1/4}$, we set them to zero.

The subproblems in Algorithm 6.5.3 are solved with the algorithm R2 [4], and we compare Algorithm 6.5.1 to TR [4] with R2 used as a subsolver, and R2 used by itself. Algorithm 6.5.3 will be denoted RIPM. Finally, we introduce a variant of RIPM named RIPMDH (*Regularized Interior Proximal Method with Diagonal Hessian approximations*), that uses the same idea as our algorithm TRDH [54]: instead of using LBFGS or LSR1 quasi-Newton approximations for $B_{k,j}$, we use diagonal quasi-Newton approximations so that (6.25) with φ defined in (6.48) can be solved analytically for specific seperable regularizers h . TRDH and RIPMDH use the Spectral Gradient update in all our results. We choose either $h(x) = \lambda\|x\|_0$ or $h(x) = \lambda\|x\|_1$, where $\lambda > 0$. When $h(x) = \lambda\|x\|_0$, RIPM and RIPMDH denote Algorithm 6.5.1 instead of

Algorithm 6.5.3 because h is not convex.

For simplicity, we described how to solve (6.1) with the constraint $x \geq 0$, but RIPM and RIPMDH are actually able to handle box constraints $\ell \leq x \leq u$. These more general constraints can be handled with minor modifications using the barrier function

$$\tilde{\phi}_k(x) := -\mu_k \sum_{i=1}^n \log(x_i - \ell_i) - \mu_k \sum_{i=1}^n \log(u_i - x_i) \quad (6.82)$$

instead of ϕ_k [34, Section 13.8]. When $\ell_i = -\infty$ (resp. $u_i = +\infty$) for some $i \in \{1, \dots, n\}$, we remove the term $\log(x_i - \ell_i)$ (resp. $\log(u_i - x_i)$) from the first (resp. second) sum in (6.82).

Our results report

- the final $f(x)$;
- the final $h(x)/\lambda$, where λ is a parameter relative to our regularization function h ;
- the final stationarity measure $\sqrt{\xi/\nu}$;
- $\|x - x^*\|$, where x_* is the exact solution, if it is available;
- the number of smooth objective evaluations $\#f$;
- the number of gradient evaluation $\#\nabla f$;
- the number of proximal operator evaluations $\#\text{prox}$;
- the elapsed time t in seconds.

Our main goal is to reduce the number of objective and gradient evaluation, as they are typically costly to evaluate. Since we did not fully optimize the allocations in our algorithms, we do not pay attention to the elapsed time, and we only report it in the tables for information.

Once a problem has been solved by all solvers, we compare their final objective values and we save the smallest, that we denote $(f + h)^*$. Then, for all solvers, we plot $(f + h)(x_k) - (f + h)^*$ for every iteration k where the gradient ∇f is evaluated. This allows us to represent the evolution of the objective per gradient evaluation. For the last gradient evaluation of RIPM and RIPMDH, we display their final objective value after applying the crossover technique.

6.6.1 Box-constrained quadratic problem

For our first numerical experiment, we solve

$$\underset{x}{\text{minimize}} \ c^T x + \frac{1}{2} x^T H x + h(x) \quad \text{subject to } \ell \leq x \leq u, \quad (6.83)$$

which is similar to [75, Section 7.1], where $h = \lambda \|\cdot\|_1$, $H = A + A^T$, $A \in \mathbb{R}^{n \times n}$ has nonzero components with probability $p = 10^{-4}$ following a normal law of mean 0 and standard deviation 1, $c \in \mathbb{R}^n$ has components generated using a normal distribution of mean 0 and

standard deviation 1, $\ell = -e - t_\ell$ and $u = e + t_u$, with $t_\ell \in \mathbb{R}^n$, $t_u \in \mathbb{R}^n$ are vectors sampled from a uniform distribution between 0 and 1. We chose $n = 10^5$, and use a LSR1 quasi-Newton approximation for TR and RIPM. For $\lambda \geq 1.0$, the components x_i of the solutions returned by TR, TRDH, R2, RIPM and RIPMDH satisfy $x_i \in \{\ell_i, u_i, 0\}$ for almost all $i \in \{1, \dots, n\}$. In this case, we observe that TR, TRDH and R2 are more efficient than RIPM. However, as we decrease λ , we get more components $x_i \notin \{\ell_i, u_i, 0\}$. We show results with $\lambda = 10^{-1}$ in Figure 6.1 and Table 6.1. TRDH performs the least amount of objective, gradient and proximal operators evaluations. RIPMDH finds the smallest final objective value, and performs fewer objective, gradient and proximal operator evaluations than TR-R2. RIPM terminates with a criticality measure higher than the other solvers, but we observe that its final objective is smaller than those of TR, TRDH and R2. For RIPM and RIPMDH, we can clearly see plateaus that delimit the outer iterations. RIPM-R2 performs many more proximal operator evaluations than RIPMDH, because it uses up to 200 R2 iterations to solve (6.25). The number of proximal operator evaluations with RIPM-R2 is also much higher than that of TR-R2, because the subproblems solved with R2 in RIPM-R2 have their objective based upon (6.48), which is not well conditioned when some components of $x_{k,j}$ approach 0, whereas the subproblems in TR-R2 are based upon (6.26).

Table 6.1 Statistics of (6.83). TR and RIPM use an LSR1 Hessian approximation. The maximum number of objective evaluations is set to 800.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\#f$	$\#\nabla f$	$\#prox$	t (s)
R2	-2.29e+04	1.5e+04	8.5e-03	679	520	679	6.8e-01
TRDH	-2.28e+04	1.5e+04	5.9e-05	57	47	113	3.6e-01
TR-R2	-2.28e+04	1.5e+04	9.9e-03	801	596	12639	8.5e+00
RIPM-R2	-2.30e+04	1.4e+04	3.4e+00	801	628	101019	5.0e+01
RIPMDH	-2.32e+04	1.5e+04	8.7e-03	313	241	628	2.7e+00

6.6.2 Sparse nonnegative matrix factorization (NNMF)

The second experiment considered is the sparse nonnegative matrix factorization (NNMF) problem from Kim and Park [52]. Let $A \in \mathbb{R}^{m \times n}$ have nonnegative entries. Each column of A represents an observation, and is generated using a mixture of Gaussians where negative entries are set to zero. We factorize $A \approx WH$ by separating A into $k < \min(m, n)$ clusters, where $W \in \mathbb{R}^{m \times k}$, $H \in \mathbb{R}^{k \times n}$ both have nonnegative entries and H is sparse. This problem can be written as

$$\underset{W, H}{\text{minimize}} \frac{1}{2} \|A - WH\|_F^2 + h(H) \quad \text{subject to } W, H \geq 0, \quad (6.84)$$

where $h(H) = \lambda \|\text{vec}(H)\|_1$ and $\text{vec}(H)$ stacks the columns of H to form a vector.

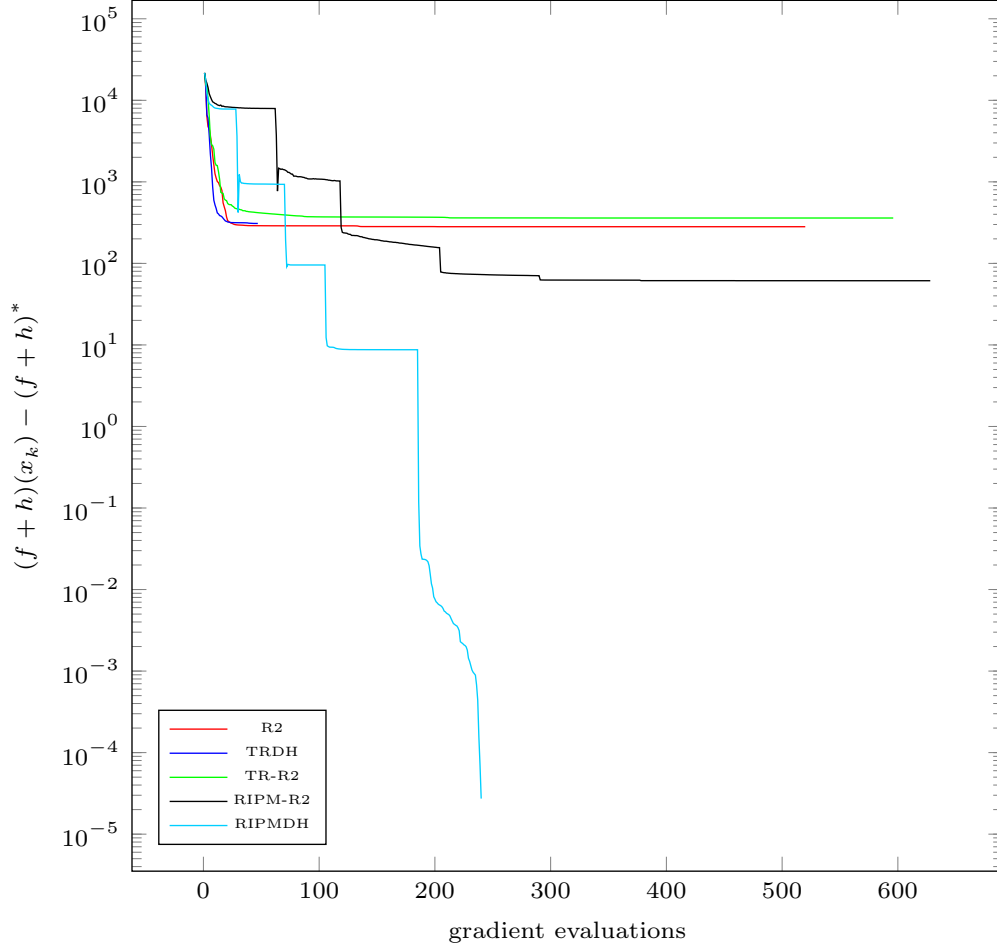


Figure 6.1 Plots of the objective of (6.83) per gradient evaluation with different solvers.

We set $m = 100$, $n = 50$, $k = 5$, $\lambda = 10^{-1}$, and report the statistics in Table 6.2. For this particular problem, we use $\epsilon_r = 10^{-6}$, which allows for more accurate solves and for a better visualization of the evolution of the objective values, shown in Figure 6.2. We observe that RIPM-R2 and RIPMDH are the only solvers to terminate. They outperform R2 and TR-R2 in terms of number of objective and gradient evaluations, and their final objective value is also smaller. R2 and TR-R2 reach the maximum number of iterations. The objective of RIPM-R2 and RIPMDH is higher than that of R2 and TR-R2 only in the early iterations, because the barrier function has more effect when μ_k is larger. RIPM-R2 performs less objective and gradient evaluations than RIPMDH, but much more proximal operator evaluations because of the reasons evoked in Section 6.6.1.

Table 6.2 Statistics of (6.84). TR and RIPM use an LSR1 Hessian approximation. The maximum number of objective evaluations is set to 8000.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\#f$	$\#\nabla f$	$\#prox$	t (s)
TRDH	1.25e+02	3.1e+01	1.6e-01	8001	6156	16000	2.0e+00
TR-R2	1.25e+02	2.8e+01	1.2e-01	8001	5122	150563	6.4e+00
RIPM-R2	1.25e+02	1.9e+01	2.5e-02	4501	3210	470975	1.1e+01
RIPMDH	1.25e+02	2.0e+01	1.4e-02	4602	3759	9205	1.3e+00

6.6.3 FitzHugh-Nagumo problem (FH)

We sample the functions $V(t; x)$ and $W(t; x)$ satisfying the FitzHugh [42] and Nagumo *et al.* [64] model for neuron activation, where $x \in \mathbb{R}^5$, as $v(x) = (v_1(x), \dots, v_{n+1}(x))$ and $w(x) = (w_1(x), \dots, w_{n+1}(x))$.

$$\frac{dV}{dt} = (V - V^3/3 - W + x_1)x_2^{-1}, \quad \frac{dW}{dt} = x_2(x_3V - x_4W + x_5). \quad (6.85)$$

The time interval $t \in [0, 20]$ is discretized, with the initial conditions $(V(0), W(0)) = (2, 0)$. We solve

$$\underset{x}{\text{minimize}} \frac{1}{2} \|(v(x) - \bar{v}(\bar{x}), w(x) - \bar{w}(\bar{x}))\|_2^2 + h(x), \quad \text{subject to } x_2 \geq 0.5, \quad (6.86)$$

where $h(x) = \lambda \|x\|_0$ with $\lambda = 10$, $n = 100$, and report the statistics in Table 6.3. Since h is not convex, we use Algorithm 6.5.1 instead of Algorithm 6.5.3. We do not show results with R2 because it encounters a numerical error during the solve of a differential equation to compute the objective. The evolution of the objective per gradient evaluation is shown in Figure 6.3. To improve readability, we choose to show the number of gradient evaluations on a logarithmic scale, and not to plot results with TRDH. All solvers converge to the value $(0.00, 0.50, 0.54, 0.00, 0.00)$ except for TRDH that has a higher final objective value than the other solvers. TR-R2 is the fastest, and seems the most suited to solve smaller problems such as (6.86). RIPM and RIPMDH still converge, but the latter is much slower. However, RIPMDH performs the least amount of proximal operator evaluations.

Table 6.3 Statistics of (6.86). TR and RIPM use an LBFGS Hessian approximation.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\#f$	$\#\nabla f$	$\#prox$	t (s)
TRDH	6.05e+00	3	2.9e+01	1001	697	2000	4.5e+00
TR-R2	4.40e+00	2	4.8e-03	53	45	4627	3.3e-01
RIPM-R2	4.40e+00	2	1.3e-02	261	112	10139	1.0e+00
RIPMDH	4.40e+00	2	1.5e-02	798	523	1600	3.8e+00

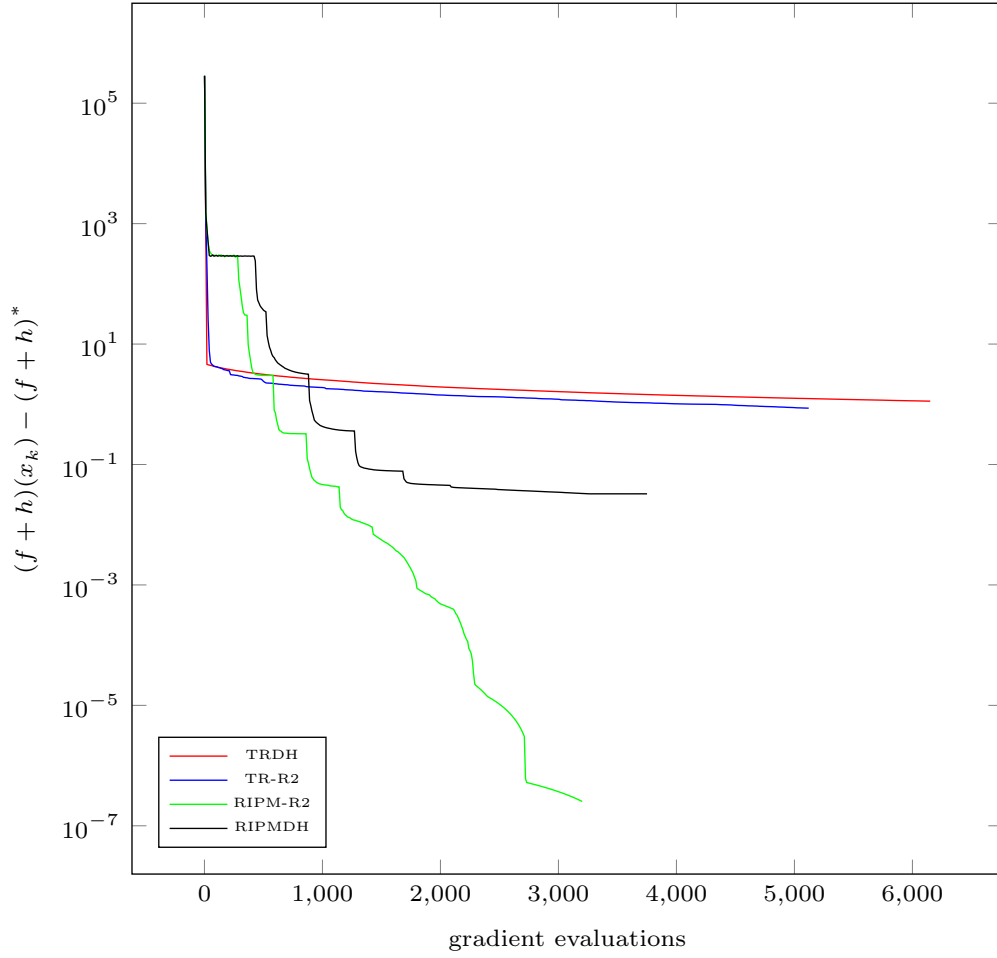


Figure 6.2 Plots of the objective of (6.84) per gradient evaluation with different solvers.

6.6.4 Constrained basis pursuit denoise (BPDN)

We solve the basis pursuit denoise problem (BPDN) [39, 78] with additional bound constraints. Let $m = 200$, $n = 512$, $b = Ax_\star + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.01)$, $A \in \mathbb{R}^{m \times n}$ has orthonormal rows, and $x_\star$ is a vector of zeros, except for 5 of its components that are set to 1. The constrained BPDN problem is written as

$$\underset{x}{\text{minimize}} \quad \frac{1}{2} \|Ax - b\|_2^2 + h(x) \quad \text{subject to } x \geq 0, \quad (6.87)$$

where $h(x) = \lambda \|x\|_1$. We use $\lambda = \|A^T b\|_\infty / 10$.

The statistics are shown in Table 6.4. R2, TRDH and TR-R2 are much more efficient than RIPM on this problem. This could come from the fact that there are many active bounds in the solution. However, this was also the case for the NNMF problem of Section 6.6.2, for which

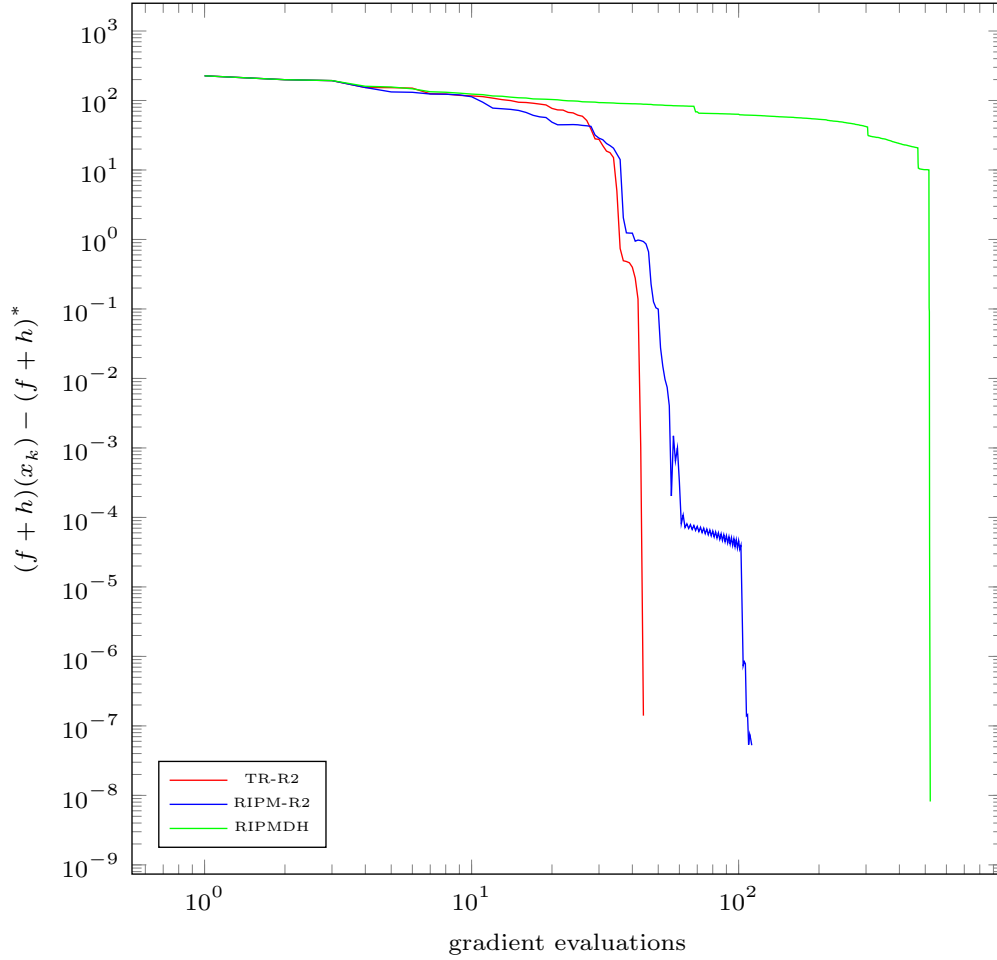


Figure 6.3 Plots of the objective of (6.86) per gradient evaluation with different solvers.

RIPM seems more efficient. Further investigations should seek to understand such behaviours on different problems. RIPM-R2-p and RIPMDH-p use the modifications $\mu_0 = 10^{-3}$ and $\epsilon_{r,i} = 1.0$, which make RIPMDH on (6.87) surpass TR-R2 and close to TRDH. Figure 6.4 shows the evolution of the objective values. RIPM and RIPMDH are not included to improve readability.

6.7 Discussion and future work

We have presented RIPM, a trust-region interior-point method to solve nonsmooth regularized problems with box constraints, and RIPMDH, a variant based upon techniques of Leconte and Orban [54]. These algorithms solve a sequence of unconstrained barrier subproblems to obtain a sequence of approximate solutions of (6.1). We have shown the convergence of the inner barrier subproblems, and we have characterized the degree of (ϵ_p, ϵ_d) -KKT optimality

Table 6.4 Statistics of (6.87). TR and RIPM use an LSR1 Hessian approximation.

solver	$f(x)$	$h(x)/\lambda$	$\sqrt{\xi/\nu}$	$\ x - x^*\ _2$	$\#f$	$\#\nabla f$	$\#prox$	t (s)
R2	3.91e-02	8.7e+00	1.8e-04	4.1e-01	11	11	11	4.0e-03
TRDH	3.91e-02	8.7e+00	6.8e-05	4.1e-01	9	9	17	8.0e-03
TR-R2	3.91e-02	8.7e+00	1.6e-04	4.1e-01	17	17	35	1.1e-02
RIPM-R2	4.12e-02	8.7e+00	7.1e-03	4.2e-01	915	915	12143	3.3e+00
RIPMDH	3.92e-02	8.7e+00	9.9e-04	4.1e-01	361	225	724	2.1e-01
RIPM-R2-p	4.22e-02	8.7e+00	1.0e-02	4.3e-01	56	56	8331	1.5e-01
RIPMDH-p	3.91e-02	8.7e+00	5.6e-04	4.1e-01	15	15	32	9.0e-03

for every outer iteration past a certain rank. Under the assumption that the iterates remain bounded, we have shown that RIPM converges to a first-order stationary point for (6.1). We compared RIPM and RIPMDH to projected-direction methods with a separable regularization function.

RIPM and RIPMDH perform well on the box-constrained quadratic problem of Section 6.6.1 and on the NMF problem of Section 6.6.2. They are not as efficient on the FH problem of Section 6.6.3 and the constrained BPDN problem of Section 6.6.4, which may suggest that projected-direction methods may be more efficient to solve problems with fewer variables and constraints. However, as observed with RIPM-R2-p and RIPMDH-p, the modification of two parameters of RIPM and RIPMDH improves their efficiency significantly on the constrained BPDN problem. This suggests that our implementation could benefit from parameter tuning.

Future work may include generalizing the algorithm to constraints of the form $c_i(x) \leq 0$ with $i \in \{1, \dots, m\}$ for some $m > 0$, where the c_i are continuously differentiable and Lipschitz-gradient continuous, as in [34, Section 13.9] in the smooth case, or [37] for nonsmooth problems.

Another improvement would be to scale the trust region to allow greater search directions along the boundary of the feasible domain. This is explained more in detail in [34, Section 13.7] for trust-regions based upon the ℓ_2 -norm. However, we could not find an alternative for trust-regions based upon the ℓ_∞ -norm that led to satisfying numerical results.

In Section 6.5.4, Model Assumption 6.5.5 does not allow the use of $h = \|\cdot\|_0$ with Algorithm 6.5.3. It would be interesting to see whether it is possible to establish convergence properties similar to those of Algorithm 6.5.1 without this assumption. One way to do this might be to replace ξ_{cp} by $\xi_{\delta_k}^{\mathcal{L}}$ in (6.35b) when j is large enough, but we did not manage to justify that this change results in a reasonable Step Assumption 6.5.1.

The extension of the convergence results to locally Lipschitz-gradient continuous functions f could also be studied, based upon the work of [36, 37, 49].

Finally, when $\{B_{k,j}\}_j$ grows unbounded, we may still be able to prove the convergence of

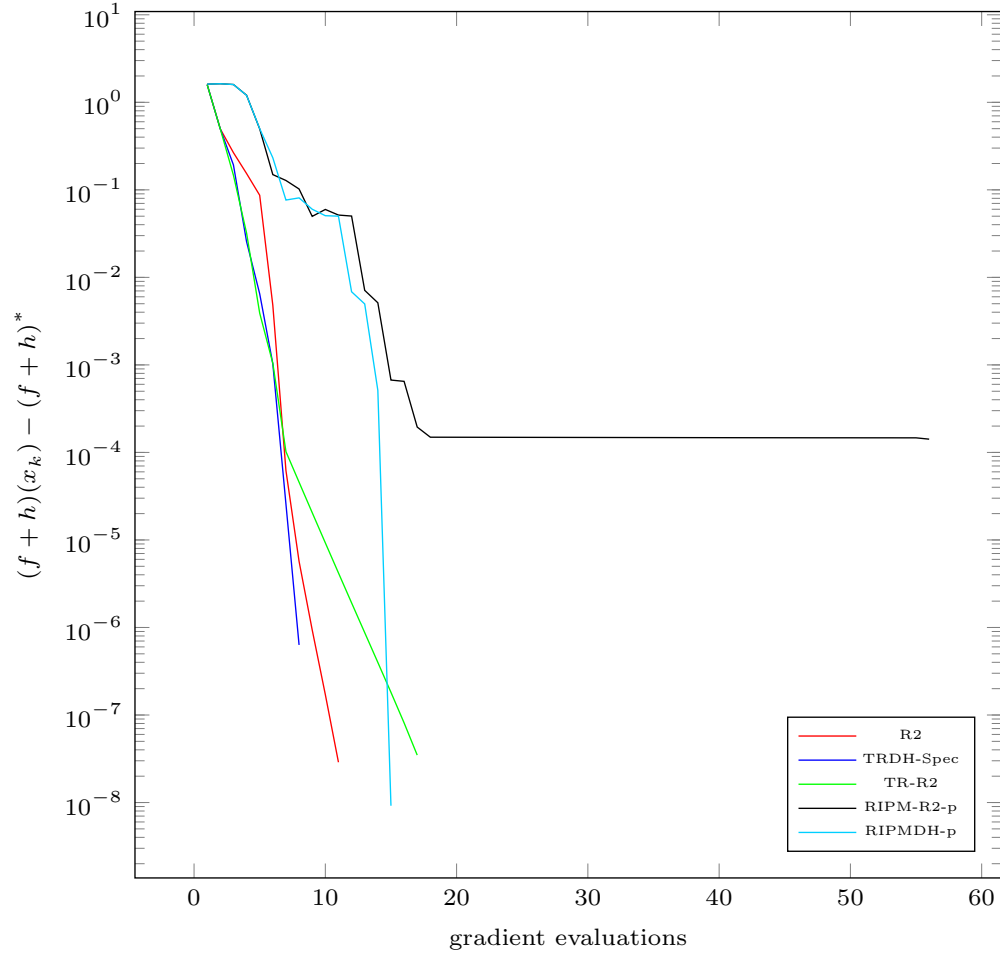


Figure 6.4 Plots of the objective of (6.87) per gradient evaluation with different solvers.

RIPM using the analysis of Leconte and Orban [55], provided that the norm of the Hessian approximations do not grow too fast.

CHAPITRE 7 CONCLUSION

7.1 Synthèse des travaux

En premier lieu, nous avons proposé une variante de l'algorithme TR [4] nommée TRDH, utilisant des approximations quasi-Newton diagonales. TRDH est capable de résoudre les sous-problèmes de région de confiance de manière analytique pour des fonctions h séparables dont on peut calculer l'opérateur proximal indéfini (*iprox* dans le chapitre 4), et bénéficie d'une borne de complexité de ϵ^{-2} similaire à celle de TR. À chaque itération, TRDH effectue l'évaluation d'un opérateur proximal (*prox*) utilisé pour calculer la mesure de criticité (de manière similaire à TR), et d'un *iprox*. Dans TR, l'évaluation du *prox* est utilisée comme point initial pour résoudre le sous-problème de région de confiance. Par contre, l'évaluation du *prox* n'est pas requise pour résoudre le sous-problème de région de confiance dans TRDH, puisqu'on dispose d'une solution analytique de ce sous-problème. Pour économiser cette évaluation, nous avons développé une variante appelée iTRDH qui évalue uniquement un *iprox* à chaque itération, et qui utilise un critère d'arrêt basé sur cet *iprox*. TRDH et iTRDH peuvent également être utilisés comme solveurs itératifs pour résoudre le sous-problème de région de confiance de TR à la place de R2 (utilisé par défaut). Sur plusieurs problèmes tests, nous avons montré qu'au moins une variante de TRDH ou iTRDH, utilisée comme solveur seul ou comme sous-solveur dans TR, était plus efficace que R2 seul ou que TR avec R2 comme sous-solveur.

Dans [4], les auteurs font l'hypothèse que les approximations quasi-Newton utilisées dans TR restent bornées. Cependant, rien n'indique que cette hypothèse est vérifiée. Nous avons donc cherché à la relâcher en modifiant légèrement TR (et TRDH) afin d'autoriser les approximations hessiennes à croître de façon non bornée, tant que leur croissance est contrôlée par un paramètre $p \in [0, 1)$. Sous cette hypothèse, nous avons réussi à montrer la convergence de l'algorithme, et à obtenir une nouvelle borne en $\epsilon^{-2/(1-p)}$. De plus, nous avons montré que cette borne est la plus fine possible en explicitant une fonction f (avec $h = 0$) mettant exactement $\lfloor \epsilon^{-2/(1-p)} \rfloor$ itérations avant de satisfaire le critère d'arrêt de TR.

Enfin, nous nous sommes intéressés à la résolution de problèmes d'optimisation non lisse avec contraintes de bornes. Pour des fonctions h séparables, nous avons vu qu'il était possible d'utiliser une version projetée de TR et de TRDH, nécessitant seulement de pouvoir projeter le *prox* et/ou l'*iprox* à l'intérieur du domaine admissible. Une alternative possible est d'utiliser une méthode de points intérieurs, utilisant une suite de sous-problèmes barrières non contraints dont l'objectif tend vers l'infini à l'approche d'une contrainte. Ces méthodes n'ont pas

forcément besoin que h soit séparable. Ainsi, nous avons implémenté une méthode de points intérieurs appelée RIPM qui résout les sous-problèmes barrière en utilisant une méthode de région de confiance inspirée de TR. Nous avons montré la convergence de RIPM sous certaines hypothèses, incluant une condition de qualification. Nous avons également développé une variante appelée RIPMDH qui utilise des approximations quasi-Newton diagonales, de manière similaire à TRDH. Nous avons effectué des comparaisons sur différents problèmes d'optimisation non lisse avec contraintes en utilisant des fonctions h séparables. RIPM et RIPMDM se comportent mieux que TR, TRDH et R2 sur les problèmes nécessitant beaucoup d'itérations avant de converger, et ont un comportement similaire sur les problèmes nécessitant moins d'itérations.

7.2 Limitations de la solution proposée

Pour les algorithmes TR, TRDH et RIPM (ainsi que leurs variantes), les résultats de convergence se basent sur l'hypothèse que ∇f est lipschitzienne. Il serait envisageable de considérer que ∇f soit localement lipschitzienne, c'est-à-dire qu'en tout point x , il existe une boule ouverte de rayon $\delta > 0$ centrée en x telle que ∇f est lipschitzienne sur cette boule.

Une autre limitation porte sur le choix des approximations quasi-Newton diagonales pour TRDH et RIPMDH. Lors de nos tests, TRDH était dans certains cas plus performant avec la méthode du gradient spectral, qui est la mise à jour la moins sophistiquée (l'approximation est un multiple de l'identité). De plus, pour RIPMDH, la méthode du gradient spectral semblait être la plus performante (et n'avons donc pas effectué de tests numériques avec les autres approximations diagonales). Ces observations pourraient mener à penser que les mises à jour PSB et Andrei présentées dans le chapitre 4 ne sont pas forcément les plus adaptées pour résoudre ce problème, bien que nous n'ayons pas d'explications sur ce comportement pour l'instant.

Pour RIPM, la preuve de convergence basée sur la mesure de criticité dépendant de $\xi_{\delta_k}^{\mathcal{L}}$ n'est pas valide si $\psi(\cdot, x)$ n'est pas convexe (dans le cas où $h(x + s) = \psi(s; x)$, si h n'est pas convexe). Certains tests que nous avons effectués avec la norme ℓ_0 mènent à penser que RIPM fonctionne quand même avec cette mesure sans l'hypothèse de convexité, mais nous n'avons pas réussi à le démontrer.

7.3 Améliorations futures

Pour finir, nous présentons quelques idées de travaux futurs. Il serait notamment intéressant de chercher à remédier à certaines difficultés soulevées dans la section 7.2. En nous inspirant

de travaux en optimisation non lisse tels que [35, 36, 49], nous pourrions étudier la convergence de nos algorithmes pour des fonctions f localement lipschitziennes. Mais dans ce cas, les bornes de complexité de nos algorithmes risquent de ne plus être valides.

Nous pourrions également chercher de nouvelles approximations quasi-Newton diagonales permettant d’obtenir de meilleures performances avec TRDH et RIPMDH. Mohamed Laghdaf Habiboullah, également doctorant à Polytechnique Montréal, étudie actuellement de telles approximations, qu’il souhaite incorporer dans le module `LinearOperators.jl`.

Nous pourrions regarder comment généraliser nos algorithmes en méthodes de région de confiance non monotones. Les méthodes non monotones en optimisation non lisse ont déjà été étudiées pour plusieurs types d’algorithmes avec recherche linéaire, voir par exemple [35, 49, 77]. À notre connaissance, il n’existe pas de méthode de région de confiance non monotone pour résoudre des problèmes non lisses, à l’exception de [82] qui considère un problème différent de (2.60).

Les solveurs implémentés effectuent certaines allocations qui pourraient être évitées. Pour TR et RIPM, il faudra implémenter les sous-solveurs qu’ils utilisent (R2, TRDH, ou la méthode du gradient proximale PG, non utilisée dans nos travaux, mais disponible dans `RegularizedOptimization.jl`) en-place, c’est-à-dire qu’il faudra pré-allouer la mémoire nécessaire au sous-solveur choisi, afin de ne pas effectuer d’allocations supplémentaires dans les itérations de TR ou RIPM. Pour cela, il faudra aussi effectuer certaines modifications dans `ShiftedProximalOperators.jl`.

Par ailleurs, nos résultats se basent uniquement sur $h = \lambda \|\cdot\|_0$ et $h = \lambda \|\cdot\|_1$, pour un paramètre $\lambda > 0$. Il serait intéressant de tester d’autres fonctions, notamment $h = \lambda \|\cdot\|_{1/2}^{1/2}$, dont l’implémentation du *prox* [83] est déjà dans `ShiftedProximalOperators.jl`, mais à laquelle il faudrait ajouter une implémentation de l’*iprox* pour pouvoir être utilisable avec TRDH et RIPMDH.

RÉFÉRENCES

- [1] N. Andrei. A diagonal quasi-Newton updating method for unconstrained optimization. *Numer. Algor.*, 81:575—590, 2019.
- [2] A. Aravkin, R. Baraldi, et D. Orban. A Levenberg-Marquardt method for nonsmooth regularized least squares. Cahier du GERAD G-2023-58, GERAD, Montréal, QC, Canada, 2022.
- [3] A. Aravkin, R. Baraldi, G. Leconte, et D. Orban. Corrigendum: A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. Cahier du GERAD G-2021-12-SM, GERAD, Montréal QC, Canada, août 2023.
- [4] A. Y. Aravkin, R. Baraldi, et D. Orban. A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. *SIAM J. Optim.*, 32(2):900–929, 2022.
- [5] H. Attouch. Convergence de fonctions convexes, des sous-différentiels et semi-groupes associés. *CR Acad. Sci. Paris*, 284(539-542):13, 1977.
- [6] H. Attouch et R. J.-B. Wets. Approximation and convergence in nonlinear optimization. In O. L. Mangasarian, R. R. Meyer, et S. M. Robinson, édés : *Nonlinear Programming 4*, p. 367–394. Academic Press, 1981. ISBN 978-0-12-468662-5.
- [7] R. Baraldi et D. P. Kouri. A proximal trust-region method for nonsmooth optimization with inexact function and gradient evaluations. *Math. Program.*, 201(1):559–598, 2022.
- [8] R. Baraldi et D. Orban. RegularizedProblems.jl: Test cases for regularized optimization. <https://github.com/JuliaSmoothOptimizers/RegularizedProblems.jl>, February 2022.
- [9] R. Baraldi et D. Orban. ShiftedProximalOperators.jl: Proximal operators for regularized optimization. <https://github.com/JuliaSmoothOptimizers/ShiftedProximalOperators.jl>, February 2022.
- [10] R. Baraldi, G. Leconte, et D. Orban. RegularizedOptimization.jl: Algorithms for regularized optimization. <https://github.com/JuliaSmoothOptimizers/RegularizedOptimization.jl>, February 2022.
- [11] A. Beck. *First-Order Methods in Optimization*. num. 25 de MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2017.
- [12] A. Beck et M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
- [13] S. Becker et J. Fadili. A quasi-Newton proximal splitting method. In F. Pereira, C. Burges, L. Bottou, et K. Weinberger, édés : *Advances in Neural Information Processing Systems*, vol. 25. Curran Associates, Inc., 2012.
- [14] S. Becker, J. Fadili, et P. Ochs. On quasi-Newton forward-backward splitting: Proximal calculus and convergence. *SIAM J. Optim.*, 29(4):2445–2481, 2019.
- [15] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, et M. Prato. Deep unfolding of a

- proximal interior point method for image restoration. *Inverse Problems*, 36(3):034005, feb 2020.
- [16] E. G. Birgin, J. M. Martínez, et M. Raydan. Spectral projected gradient methods: Review and perspectives. *Journal of Statistical Software*, 60(3):1–21, 2014.
 - [17] J. Bolte, S. Sabach, et M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, (146):459–494, 2014.
 - [18] S. Boyd, N. Parikh, E. Chu, B. Peleato, et J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
 - [19] S. P. Boyd et L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
 - [20] R. I. Boş, E. R. Csetnek, et S. C. László. An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. *EURO J. Comput. Optim.*, 4(1):3–25, 2016.
 - [21] R. H. Byrd, M. E. Hribar, et J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM J. Optim.*, 9(4):877–900, 1999.
 - [22] R. H. Byrd, J. C. Gilbert, et J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. 89(1):149–185, 2000.
 - [23] W. Cao, J. Sun, et Z. Xu. Fast image deconvolution using closed-form thresholding formulas of L_q ($q = \frac{1}{2}, \frac{2}{3}$) regularization. *J. Vis. Commun. Image R.*, 24(1):31–41, 2013.
 - [24] R. G. Carter. Safeguarding hessian approximations in trust region algorithms. Rap. tech. TR87-12, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA, 1987.
 - [25] C. Cartis, N. I. M. Gould, et Ph. L. Toint. On the complexity of steepest descent, Newton’s and regularized Newton’s methods for nonconvex unconstrained optimization problems. *SIAM J. Optim.*, 20(6):2833–2852, 2010.
 - [26] C. Cartis, N. I. M. Gould, et Ph. L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM J. Optim.*, 21(4):1721–1739, 2011.
 - [27] C. Cartis, N. I. M. Gould, et Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: Motivation, convergence and numerical results. *Math. Program.*, 127(2):245–295, 2011.
 - [28] C. Cartis, N. Gould, et P. L. Toint. Strong evaluation complexity bounds for arbitrary-order optimization of nonconvex nonsmooth composite functions, 2020.
 - [29] C. Cartis, N. I. M. Gould, et Ph. L. Toint. Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints. *SIAM J. Optim.*, 30(1):513–541, 2020.
 - [30] C. Cartis, N. I. M. Gould, et Ph. L. Toint. *Evaluation Complexity of Algorithms for Nonconvex Optimization*. num. 30 de MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2022.
 - [31] Z. Chen, A. Milzarek, et Z. Wen. A trust-region method for nonsmooth nonconvex optimization,

- 2021.
- [32] E. Chouzenoux, M.-C. Corbineau, et J.-C. Pesquet. A Proximal Interior Point Algorithm with Applications to Image Processing. 62(6):919–940.
 - [33] A. R. Conn, N. I. M. Gould, D. Orban, et Ph. L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. 87(2):215–249, 2000.
 - [34] A. R. Conn, N. I. M. Gould, et Ph. L. Toint. *Trust-Region Methods*. num. 1 de MOS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.
 - [35] A. De Marchi. Proximal gradient methods beyond monotony. *Journal of Nonsmooth Analysis and Optimization*, Volume 4, juin 2023.
 - [36] A. De Marchi et A. Themelis. Proximal Gradient Algorithms Under Local Lipschitz Gradient Continuity. *J. Optim. Theory and Applics.*, 194(3):771–794, 2022.
 - [37] A. De Marchi et A. Themelis. An interior proximal gradient method for nonconvex optimization, 2024.
 - [38] J. E. Dennis, Jr. et H. Wolkowicz. Sizing and least-change secant methods. *SIAM J. Numer. Anal.*, 30(5):1291–1314, 1993.
 - [39] D. Donoho. Compressed sensing. *IEEE T. Inform. Theory*, 52(4):1289–1306, 2006.
 - [40] J. Duchi, E. Hazan, et Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011.
 - [41] J.-P. Dussault, T. Migot, et D. Orban. Scalable adaptive cubic regularization methods. *Math. Program.*, 2023.
 - [42] R. FitzHugh. Mathematical models of threshold phenomena in the nerve membrane. *B. Math. Biophys.*, 17(4):257–278, 1955.
 - [43] R. Fletcher. *A model algorithm for composite nondifferentiable optimization problems*, p. 67–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982. DOI: 10.1007/BFb0120959.
 - [44] M. Fukushima et H. MINE. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8):989–1000, 1981.
 - [45] D. M. Gay. Computing optimal locally constrained steps. *SIAM J. Sci. and Statist. Comput.*, 2(2):186–197, 1981.
 - [46] J.-C. Gilbert et C. Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Math. Program.*, 45:407–435, 1989.
 - [47] A. A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70(5):709 – 710, 1964.
 - [48] C. Kanzow et T. Lechner. Globalized inexact proximal Newton-type methods for nonconvex composite functions. *Comput. Optim. Appl.*, 78(2):377–410, 2021.
 - [49] C. Kanzow et P. Mehlitz. Convergence Properties of Monotone and Nonmonotone Proximal Gradient Methods Revisited. *J. Optim. Theory and Applics.*, 195(2):624–646, 2022.
 - [50] H. Karimi, J. Nutini, et M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In P. Frasconi, N. Landwehr, G. Manco, et

- J. Vreeken, éd. : *Machine Learning and Knowledge Discovery in Databases*, p. 795–811, Cham, 2016. Springer International Publishing.
- [51] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.
 - [52] J. Kim et H. Park. Sparse nonnegative matrix factorization for clustering. Rap. tech. GT-CSE-08-01, Georgia Inst. of Technology, 2008.
 - [53] H. Y. Le. Generalized subdifferentials of the rank function. *Optimization Letters*, 7(4):731–743, 2013.
 - [54] G. Leconte et D. Orban. The indefinite proximal gradient method. Cahier du GERAD G-2023-37, GERAD, Montréal QC, Canada, août 2023.
 - [55] G. Leconte et D. Orban. Complexity of trust-region methods with unbounded Hessian approximations for smooth and nonsmooth optimization. Les Cahiers du GERAD G-2023-65, Groupe d’études et de recherche en analyse des décisions, GERAD, Montréal QC H3T 2A7, Canada, déc. 2023.
 - [56] G. Leconte et D. Orban. An interior-point trust-region method for nonsmooth regularized bound-constrained optimization. Cahier du GERAD G-2024-17, GERAD, Montréal QC, Canada, fév. 2024.
 - [57] C.-p. Lee et S. J. Wright. Inexact successive quadratic approximation for regularized optimization. *Comput. Optim. Appl.*, 72:641–674, 2019.
 - [58] J. D. Lee, Y. Sun, et M. A. Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM J. Optim.*, 24(3):1420–1443, 2014.
 - [59] E. Levitin et B. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6:1–50, 12 1966.
 - [60] H. Li et Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, et R. Garnett, éd. : *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015.
 - [61] P.-L. Lions et B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.
 - [62] S. Lotfi, T. Bonniot de Ruisselet, D. Orban, et A. Lodi. Stochastic damped L-BFGS with controlled norm of the Hessian approximation. 2020. OPT2020 Conference on Optimization for Machine Learning.
 - [63] J. J. Moré et D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. and Statist. Comput.*, 4(3):553–572, 1983.
 - [64] J. Nagumo, S. Arimoto, et S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.
 - [65] J. L. Nazareth. If quasi-Newton then why not quasi-Cauchy ? endif. *SIAG/OPT Views-and-News*, 6:11–14, 1995.
 - [66] Y. Nesterov. A method for solving the convex programming problem with convergence rate

- $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.
- [67] Y. Nesterov et B. Polyak. Cubic regularization of Newton method and its global performance. 108(1):177–205.
 - [68] J. Nocedal et S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, 2nd édn, 2006.
 - [69] P. Ochs, Y. Chen, T. Brox, et T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM J. Imaging Sci.*, 7(2):1388–1419, 2014.
 - [70] D. Orban, A. S. Siqueira, et contributors. LinearOperators.jl. <https://github.com/JuliaSmoothOptimizers/LinearOperators.jl>, September 2020.
 - [71] N. Parikh et S. Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, jan 2014.
 - [72] R. A. Poliquin. An extension of Attouch’s theorem and its application to second-order epidifferentiation of convexly composite functions. *Transactions of the American Mathematical Society*, 332:861–874, 1992.
 - [73] R. Rockafellar et R. Wets. *Variational Analysis*, vol. 317. Springer Verlag, 1998.
 - [74] K. Scheinberg et X. Tang. Practical inexact proximal quasi-Newton method with global complexity analysis. *Math. Program.*, (160):495–529, 2016.
 - [75] C. Shen, W. Xue, L.-H. Zhang, et B. Wang. An Active-Set Proximal-Newton Algorithm for ℓ_1 Regularized Optimization Problems with Box Constraints. *Journal of Scientific Computing*, 85(3):57, 2020.
 - [76] L. Stella, A. Themelis, P. Sopasakis, et P. Patrinos. A simple and efficient algorithm for nonlinear model predictive control. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, p. 1939–1944, 2017.
 - [77] A. Themelis, L. Stella, et P. Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms. *SIAM J. Optim.*, 28(3):2274–2303, 2018.
 - [78] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.
 - [79] Ph. L. Toint. Global convergence of a class of trust-region methods for nonconvex minimization in hilbert space. *IMA J. Numer. Anal.*, 8(2):231–252, 04 1988.
 - [80] B. van der Pol. LXXXVIII. On “relaxation-oscillations”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):978–992, 1926.
 - [81] M. H. Wright. Interior methods for constrained optimization. *Acta Numer.*, 1:341–407, 1992.
 - [82] M. Xi et A. Xiao. A nonmonotone trust region method for nonsmooth composite programming problems. *American Journal of Applied Mathematics*, 6(2):71–77, 2018.
 - [83] Z. Xu, X. Chang, F. Xu, et H. Zhang. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1013–1027, 2012.
 - [84] S. Yagishita et S. Nakayama. An acceleration of proximal diagonal newton method. *JSIAM*

Letters, 16:5–8, 2024.

- [85] H. Yamashita, H. Yabe, et T. Tanabe. A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization. *Math. Program.*, 102(1):111–151, 2005.
- [86] M. Zhu, J. L. Nazareth, et H. Wolkowicz. The quasi-Cauchy relation and diagonal updating. *SIAM J. Optim.*, 9(4):1192–1204, 1999.