Titre: Title:	Méthodes mises à l'échelle pour la reconstruction tomographique en coordonnées cylindriques
Auteur: Author:	Guillaume Mestdagh
Date:	2019
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Mestdagh, G. (2019). Méthodes mises à l'échelle pour la reconstruction tomographique en coordonnées cylindriques [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/4050/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/4050/
Directeurs de recherche: Advisors:	Yves Goussard, & Dominique Orban
Programme: Program:	génie électrique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Méthodes mises à l'échelle pour la reconstruction tomographique en coordonnées cylindriques

GUILLAUME MESTDAGH
Département de génie électrique

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

Août 2019

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Méthodes mises à l'échelle pour la reconstruction tomographique en coordonnées cylindriques

présenté par **Guillaume MESTDAGH**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Jean-François FRIGON, président
Yves GOUSSARD, membre et directeur de recherche
Dominique ORBAN, membre et codirecteur de recherche
Jean-Pierre DUSSAULT, membre

REMERCIEMENTS

J'aimerais remercier plusieurs personnes qui ont joué un rôle important pendant ces deux années de maîtrise.

Tout d'abord, je remercie mon directeur Yves Goussard pour m'avoir permis d'effectuer ma maîtrise dans de bonnes conditions matérielles. Merci également à mon co-directeur Dominique Orban pour son implication dans le projet et notamment dans la rédaction de l'article, pour m'avoir transmis son goût pour l'optimisation, pour m'avoir fait participer aux Journées de l'optimisation. Merci enfin à Jean-François Frigon et Jean-Pierre Dussault qui ont accepté de participer à mon jury.

Pour avoir rendu ces deux années plus agréables, je remercie mes collègues de travail, et en particulier Mélanie, Nicolas, Aurélien, Thiago, Jean-Baptiste, Alexis et Tiphaine, qui ont été bien plus que des collègues.

Finalement, je salue mes colocataires de la rue Saint-Urbain, les musiciens des *Van Hornies* et tous ceux qui m'ont accompagné dans les multiples aventures qui ont jalonné mon séjour à Montréal.

RÉSUMÉ

Nous traitons le problème de reconstruction d'image en tomographie à rayons X. La reconstruction d'une image par une approche statistique requiert la résolution d'un problème d'optimisation convexe de grande taille à contraintes de bornes. La discrétisation de l'image en coordonnées cylindriques permet d'économiser de la mémoire et de réduire le volume de calculs, mais dégrade significativement le conditionnement du problème de reconstruction.

Pour mettre le problème à l'échelle, nous disposons d'un préconditionneur basé sur la structure bloc-circulante de la matrice du système qui apparaît en coordonnées cylindriques. La mise à l'échelle améliore considérablement la vitesse de reconstruction, à condition qu'on l'applique de manière appropriée. En particulier, on cherche à préserver les contraintes de bornes, car elles donnent lieu à des projections orthogonales peu coûteuses.

Nous traitons le problème de reconstruction avec des méthodes d'optimisation à directions projetées, pour lesquelles on applique la mise à l'échelle aux directions de descente. Alors qu'en imagerie la mise à l'échelle est souvent diagonale, nous proposons une stratégie permettant d'utiliser notre préconditionnement non-diagonal dans le cadre des contraintes de bornes. Celle-ci se base sur des opérateurs de mise à l'échelle partiellement diagonaux définis en fonction des contraintes actives à l'itéré courant. Nous appliquons la mise à l'échelle à deux algorithmes, TRON, une méthode de Newton à région de confiance, et L-BFGS-B, une méthode de quasi-Newton à mémoire limitée et à recherche linéaire.

Nous comparons nos méthodes à une approche précédente dans laquelle le problème est transformé par changement de variable. Les essais numériques réalisés pour des problèmes de reconstruction d'image montrent que notre approche nécessite un plus faible volume de calcul que celle avec changement de variable et rend la reconstruction plus rapide. Par ailleurs, l'usage de méthodes d'ordre supérieur permet de résoudre le problème avec une plus grande précision qu'une méthode de premier ordre.

ABSTRACT

We consider the image reconstruction problem in X-Ray computed tomography. Image reconstruction by a statistical approach yields large-scale convex optimization problems with bound contraints. Discretizing the image in cylindrical coordinates results in savings in terms of memory and computations, but also in a badly scaled reconstruction problem.

We scale the problem by using a matrix based on the bloc-circulant structure of the system matrix in cylindrical coordinates. In order to improve the reconstruction speed, we need to apply the scaling matrix wisely. In particular, we should preserve the bound constraints, because they yield cheap orthogonal projections.

We solve the reconstruction problem with projected-directions optimization methods where the scaling is applied to descent directions. While the scaling is usually diagonal in imaging applications, we use a strategy to handle our nondiagonal scaling in the context of bound-constrained problems. The strategy involves the use of partially diagonal scaling operators, which depend on the active constraints at the current iterate. We describe our implementation of the scaling strategy into two algorithms: TRON, a trust-region method with exact second derivatives, and L-BFGS-B, a linesearch method with a limited-memory quasi-Newton Hessian approximation.

We compare our approach with a previous one where a change of variable is made in the problem. Numerical tests carried on two reconstruction problems show that our approach gives superior results in term of computational time than the change of variable approach, and achieves much tighter accuracy than a first-order method.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

BFGS	Méthode de Broyden, Fletcher, Goldfarb et Shanno
CG	*Conjugate Gradient*, méthode du gradient conjugué
CT	*Computed Tomography*, tomographie numérique
FISTA	*Fast Iterative Shrinkage-Thresholding Algorithm*, méthode de gradient proximal accélérée
L-BFGS	*Limited-memory BFGS*, méthode BFGS à mémoire limitée
L-BFGS-B	*L-BFGS for Bound contraints*, méthode L-BFGS pour contraintes de bornes
OS-SQS	*Ordered Subsets - Separable Quadratic Surrogates*, méthode de sous-ensembles ordonnés à fonction substitut quadratique séparable
SPG	*Spectral Projected Gradient*, méthode du gradient projeté spectral
TRON	*Trust-Region Newton method*, méthode de Newton à région de confiance

CHAPITRE 1 INTRODUCTION

1.1 Définition et concepts de base

Dans ce mémoire, nous nous intéressons à la résolution d'un problème de reconstruction tomographique au moyen de méthodes de directions projetées. Dans cette section, nous présentons d'abord rapidement la tomographie par rayons X, puis nous effectuons quelques rappels sur le cadre mathématiques des méthodes à directions projetées.

1.1.1 Tomographie par rayons X

La tomographie par rayons X consiste à produire une image en trois dimensions de l'intérieur d'un patient, à partir d'une série de mesures de transmission réalisées selon plusieurs directions. Pour chaque direction, on mesure l'atténuation d'un faisceau de rayons qui traverse le patient. L'image reconstruite correspond au coefficient d'atténuation des rayons X à chaque point du patient (Herman, 2009).

L'acquisition de données se fait à l'aide d'un tomographe, dont on illustre la géométrie à la figure 1.1. La source et les détecteurs tournent autour du patient pour réaliser des mesures sous différents angles (Herman, 2009). On recueille ainsi suffisamment d'information pour reconstruire une carte de densité du patient à l'aide d'une méthode de reconstruction.

L'exposition aux rayons X étant nocive pour la santé, la diminution des doses de rayonnement infligées au patient est un axe de recherche actuel en tomographie médicale. À cette préoccupation sanitaire s'ajoute un besoin d'images toujours plus précises, pouvant être traitées automatiquement ou contenant des détails plus fins.

Les méthodes de reconstruction statistiques offrent une réponse à ces deux enjeux antagonistes (Fessler, 2000). Elles sont basées sur un modèle physique qui prend en compte l'aspect aléatoire des mesures d'atténuation. Elles s'opposent aux méthodes analytiques comme la rétroprojection filtrée (Feldkamp, Davis, et Kress, 1984), et sont connues pour donner des images de meilleure qualité que ces dernières.

Ravishankar, Ye, et Fessler (2019) dressent un bref historique des tendances en reconstruction tomographique, depuis les premières méthodes directes jusqu'à l'apprentissage machine. On pourra aussi consulter Geyer, Schoepf, Meinel, Nance Jr, Bastarrika, Leipsic, Paul, Rengo, Laghi, et De Cecco (2015) pour un aperçu des méthodes de reconstruction itératives implantées dans les tomographes commerciaux.

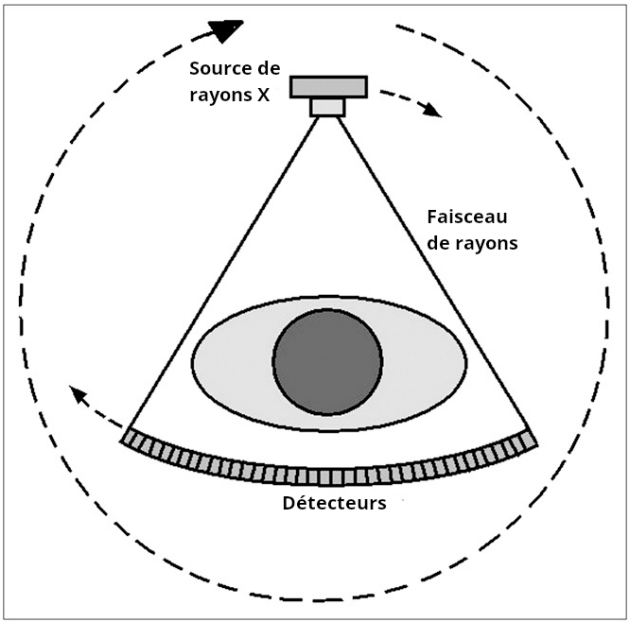

Figure 1.1 Géométrie d'un tomographe médical (Goldman, 2007). Le patient est au centre du dispositif et l'ensemble formé par la source et les détecteurs tourne autour de lui.

1.1.2 Modèle direct et problème de reconstruction

La reconstruction par une approche statistique repose sur un modèle direct de formation des données. On le décrit brièvement. Dans son mémoire, McLaughlin (2017) présente plus en détail le modèle utilisé.

On représente le patient par une distribution de coefficients d'atténuation sur le domaine D notée $x : D \to \mathbb{R}^+$. En pratique, on partitionne D en n sous-ensembles appelés pixels (ou parfois voxels). L'image est alors représentée par le vecteur x qui contient la valeur de l'atténuation sur chaque pixel.

Les données mesurées sont représentées par un vecteur b. Chaque composante b_i est liée au nombre de photons I_i transmis pour un angle de mesure et pour un détecteur du tomographe. On a $b_i = \ln(I_i/I_0)$, où I_0 est l'intensité de la source. La mesure $I_i \in \mathbb{N}$ est entachée d'un bruit qui suit une distribution de Poisson.

Le modèle direct pour la formation des données est de type

$$b = Ax + n, \tag{1.1}$$

où n est un terme de bruit et A est la matrice du système. Les coefficients de A correspondent aux intersections entre les rayons allant de la source au détecteur et les pixels de l'image pour

chaque mesure. Plus précisément, lors de la i-ième mesure, le rayon traverse le j-ième pixel sur une longueur A_{ij}. Comme un rayon ne traverse que quelques pixels, A est très creuse.

A l'aide d'une approche par maximum a posteriori, Sauer et Bouman (1993) estiment x en résolvant le problème de moindres carrés régularisé

$$\min_{x \geqslant 0} \|Ax - b\|_V^2 + R(x), \tag{1.2}$$

où $R : \mathbb{R}^n \to \mathbb{R}$ est une fonction de régularisation convexe, et V est une matrice de pondération diagonale de terme général $V_{ii} = I_i/I_0$.

La fonction R limite le bruit sur l'image en pénalisant les variations trop brusques de l'image. Pour la reconstruction, nous utilisons la pénalisation L_2/L_1

$$R(x) = \lambda \sum_k \sqrt{\delta^2 + [Kx]_k^2}, \tag{1.3}$$

où K est une matrice de différences finies et $\lambda, \delta > 0$. La fonction R choisie croît linéairement pour les grandes différences entre pixels, préservant ainsi les bords des images. C'est une approximation différentiable de la pénalisation $\|Kx\|_1$. Zhang, Wang, Zeng, Tao, et Ma (2018) proposent un aperçu des différentes stratégies de pénalisation en tomographie à rayons X.

1.1.3 Contraintes et projections

Le problème de reconstruction (1.2) est un problème d'optimisation de la forme

$$\min_{x \geqslant 0} f(x), \tag{1.4}$$

où $x \in \mathbb{R}^n$ et $f : \mathbb{R}^n \mapsto \mathbb{R}$ est convexe et de classe $\mathcal{C}^2$.

Ici les contraintes s'appliquent indépendemment à chaque composante du vecteur x. On associe chaque contrainte à la composante de x correspondante, c'est-à-dire que la i-ième contrainte est $x_i \geqslant 0$.

Pour mieux comprendre les méthodes à directions projetées, nous rappelons quelques notions liées à l'optimisation sous contraintes de bornes. On trouvera plus d'informations sur l'optimisation en général dans les ouvrages de référence de Boyd et Vandenberghe (2004) et Nocedal et Wright (2006).

Conditions d'optimalité et contraintes actives

Dans le problème (1.4), l'ensemble des solutions réalisables est $\mathbb{R}^n_+ = \{w \in \mathbb{R}^n \,|\, w \geqslant 0\}$. Cet ensemble est convexe. Comme la fonction objectif est convexe aussi, la recherche des solutions se résume à trouver les points stationnaires de (1.4).

Nous ne débattons pas ici de l'existence et l'unicité des solutions. Dans l'application, la fonction objectif est strictement convexe et coercive. Il y a donc une unique solution caractérisée par les conditions d'optimalité.

On écrit les conditions d'optimalité pour (1.4). Le point x^* est un point stationnaire de f sur $\mathbb{R}^n_+$ quand

$$\begin{cases} \nabla f(x^*) - \lambda^* = 0 \\ \forall i \quad x_i^* \geqslant 0 \\ \forall i \quad \lambda_i^* \geqslant 0 \\ \forall i \quad \lambda_i^* = 0 \text{ ou } x_i^* = 0, \end{cases} \tag{1.5}$$

où λ^* est un vecteur de multiplicateurs de Lagrange.

Ainsi, pour chaque composante de x^*, deux cas sont possibles.

- Si $\lambda_i^* > 0$, alors $[\nabla f(x^*)]_i > 0$ et $x_i^* = 0$; la i-ième contrainte est dite active, car c'est elle qui empêche f d'atteindre des valeurs plus faibles.
- Si $x_i^* > 0$, alors $[\nabla f(x^*)]_i = 0$ et la contrainte est dite inactive.

Dans chaque direction, la fonction f ne peut plus décroître au premier ordre, soit parce que sa dérivée est nulle dans cette direction, soit parce que les contraintes ne permettent pas de se déplacer dans cette direction.

Faces de $\mathbb{R}^n_+$ et ensemble actif

L'ensemble $\mathbb{R}^n_+$ est un polyèdre. À tout ensemble d'indice $\mathcal{A} \subset [\![1, n]\!]$, on associe la face de $\mathbb{R}^n_+$

$$F_+(\mathcal{A}) = \{x \in \mathbb{R}^n_+ \,|\, \forall i \in \mathcal{A} \quad x_i = 0\}\,, \tag{1.6}$$

et le sous-espace engendré par $F_+(\mathcal{A})$

$$F(\mathcal{A}) = \{x \in \mathbb{R}^n \,|\, \forall i \in \mathcal{A} \quad x_i = 0\}\,. \tag{1.7}$$

Ainsi, les sous-espaces de la forme $F(\mathcal{A})$ sont associés à un ensemble de contraintes pour lesquelles l'égalité est respectée.

On définit la face active à partir de la notion de contraintes actives. Pour un point réalisable x,

l'ensemble actif

$$I_+(x) = \{i \mid x_i = 0 \text{ et } [\nabla f(x)]_i > 0\}, \tag{1.8}$$

correspond aux contraintes que l'on viole si on se déplace dans la direction $-\nabla f(x)$ en partant de x. On lui associe $F(x) = F(I_+(x))$ ainsi que la face active $F_+(x) = F_+(I_+(x))$. La face active est la face par laquelle on sort de l'ensemble réalisable si on suit la direction $-\nabla f(x)$ depuis le point x.

La notion de face active est fondamentale dans les méthodes de directions projetées. Le but est d'identifier l'ensemble actif $I_+(x^*)$ à la solution et de calculer cette solution en résolvant un problème sans contraintes restreint à $F(x^*)$. Burke et Moré (1994) donnent des détails sur l'identification de la face active dans le cas plus général où l'ensemble réalisable est un polyèdre.

Projection orthogonale

Qui dit méthodes de directions projetées dit projections orthogonales sur l'ensemble réalisable. L'opération de projection n'a pas le même sens selon qu'on projette un point ou une direction.

Projeter un point $x \in \mathbb{R}^n$ sur le convexe fermé $\Omega \subset \mathbb{R}^n$, c'est trouver le point réalisable v le plus proche de x. Le point v est donc solution v du problème

$$\min_{v \in \Omega} \quad \|x - v\|. \tag{1.9}$$

Selon les caractéristiques de Ω et le choix de la norme $\|\cdot\|$, la projection est plus ou moins coûteuse. Si $\Omega = \mathbb{R}^n_+$ et $\|\cdot\| = \|\cdot\|_2$, on calcule la projection facilement par la formule directe

$$\text{Proj}(x) = \max(x, 0), \tag{1.10}$$

où le maximum s'applique élément par élément. Pour cette raison il est commode d'utiliser des méthodes de directions projetées pour des problèmes à contraintes de bornes.

La projection d'une direction est légèrement différente. Considérons le chemin projeté

$$v(\alpha) = \text{Proj}(x + \alpha d), \tag{1.11}$$

où $x \in \Omega$ et $d \in \mathbb{R}^n$. Le chemin projeté (1.11) est la projection point par point du chemin $x + \alpha d$. La direction projetée d_{Proj} est la direction selon laquelle on se dirige effectivement quand on suit le chemin projeté (1.11) en partant de x. Autrement dit, $d_{\text{Proj}} = \dot{v}(0)$.

En particulier, si Ω est un polyèdre, la direction projetée d_{Proj} et le vecteur directeur du premier segment du chemin projeté (1.11). Ce cas est illustré sur la figure 1.2.

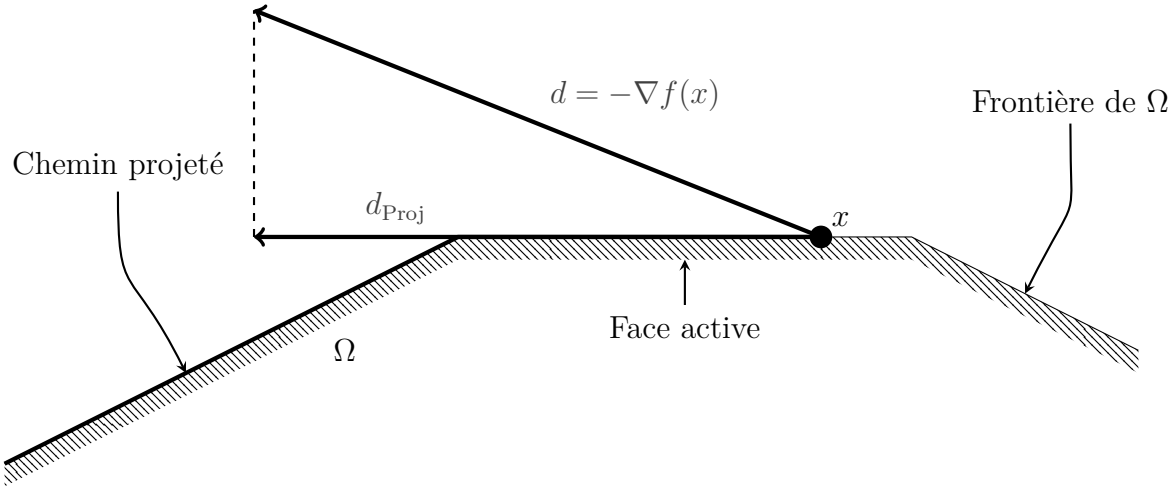

Figure 1.2 Direction projetée et chemin projeté point par point dans le cas où Ω est un polyèdre. La direction du premier segment du chemin projeté est la projection de d sur le sous-espace engendré par la face active.

Plus précisément, d_{Proj} est la projection orthogonale de d sur le cône des directions admissibles (Nocedal et Wright, 2006, définition 12.2). Dans le cas $\Omega = \mathbb{R}^n_+$, le cône des directions admissibles a pour expression

$$\mathcal{K} = \{w \in \mathbb{R}^n \mid \forall i \quad x_i = 0 \Rightarrow w_i \geqslant 0\}. \tag{1.12}$$

Pour projeter d, on identifie d'abord l'ensemble des contraintes qui seraient violées par un pas selon d, puis on met les coordonnées correspondantes de la direction projetée w à zéro. On obtient

$$[d_{\text{Proj}}]_i = \begin{cases} 0 & \text{si } x_i = 0 \text{ et } d_i < 0 \\ d_i & \text{sinon.} \end{cases} \tag{1.13}$$

La définition (1.13) fait penser à la définition de l'ensemble actif (1.8). Souvent, la direction à projeter est $d = -\nabla f(x)$, et dans ce cas on projette la direction sur le sous-espace $F(x)$. C'est ce qu'on fait sur la figure 1.2.

Les méthode de directions projetées utilisent massivement les projections de points comme de directions.

1.2 Éléments de la problématique

Dans cette section nous présentons le projet dans lequel s'inscrit ce travail, ainsi que les travaux précédents qui ont donné lieu à ce sujet.

1.2.1 Coordonnées cylindriques

La particularité de ce projet réside dans le choix de la forme des pixels de l'image. Alors que les images sont classiquement découpées en voxels cubiques, on discrétise l'image en coordonnées cylindriques (r, θ, z) (Thibaudeau, Leroux, Fontaine, et Lecomte, 2013). L'angle de décalage $\Delta\theta$ entre deux secteurs voisins est égal à la différence entre deux angles de mesure.

Or, on a vu que les propriétés numériques du problème (1.2) dépendent fortement de la manière dont on discrétise l'image. En particulier, A et K sont définies à partir de la discrétisation choisie.

L'usage de coordonnées cylindriques se traduit par une structure bloc-circulante pour la matrice du système A. L'espace mémoire nécessaire pour stocker A est donc considérablement réduit, car il ne faut stocker qu'une rangée de blocs (Goussard, Golkar, Wagner, et Voorons, 2013). On note que la matrice de différences finies K est également bloc-circulante.

Cependant, l'existence de pixels de tailles différentes dégrade le conditionnement de A. Les difficultés rencontrées par les solveurs itératifs pour résoudre (1.2) en coordonnées cylindriques se reflètent par l'apparition d'artefacts au centre des images.

1.2.2 Matrice de mise à l'échelle

Afin d'améliorer les propriétés numériques de (1.2), Golkar (2013) propose une matrice de mise à l'échelle qui tire profit de la structure circulante par blocs de A et K.

On note f la fonction objectif de (1.2). Sa hessienne s'écrit

$$\nabla^2 f(x) = A^{\mathrm{T}} V A + \lambda K^{\mathrm{T}} \mathrm{diag}(r) K \quad \text{où} \quad r_i = \frac{\delta^2}{\left(\delta^2 + [Kx]_i^2\right)^{3/2}}. \tag{1.14}$$

On commence par créer une approximation bloc-circulante de la hessienne. Tout d'abord, pour $x = 0$, on a $\nabla^2 R(0) = \frac{\lambda}{\delta} K^{\mathrm{T}} K$, qui est une matrice bloc-circulante. De plus on remplace V par l'identité pour avoir un premier terme bloc-circulant. La matrice bloc-circulante résultante est bloc-diagonalisée à l'aide d'une transformée de Fourier par blocs (Chen, 2005). On a

$$A^{\mathrm{T}}A + \frac{\lambda}{\delta}K^{\mathrm{T}}K = F^*HF, \tag{1.15}$$

où H est diagonale par blocs. Afin d'avoir $C^{\mathrm{T}}\nabla^2 f(0)C \approx I$, on définit la matrice de mise à l'échelle par $C = F^*T^{-1/2}F$ où $T = \mathrm{diag}(H)$. La matrice C et son inverse peuvent être appliquées au coût modique de $\mathrm{O}(n \ln n)$ opérations. C'est pourquoi on dit parfois que C est un opérateur rapide.

En effectuant le changement de variable $x = Cu$, on obtient le problème mis à l'échelle

$$\min_{Cu \geqslant 0} \|ACu - b\|_V^2 + R(Cu), \tag{1.16}$$

où R est définie en (1.3). Ce problème a une hessienne mieux conditionnée que (1.2), et donne lieu à une convergence plus rapide avec des solveurs itératifs, comme on le voit en section 4.3. Cependant, la transformation des contraintes de bornes en contraintes affines ajoute une difficulté au traitement du problème. En effet, on ne peut plus utiliser de formule directe pour effectuer des projections orthogonales sur l'ensemble réalisable.

McLaughlin (2017) traite (1.16) à l'aide d'une méthode de Newton à directions projetées adaptée de TRON (Lin et Moré, 1999). McLaughlin calcule efficacement les projections orthogonales en résolvant itérativement un problème de moindres carrés linéaires dont l'opérateur est C et avec contraintes de bornes.

Entre placer certaines coordonnées d'un vecteur à zéro et résoudre itérativement un problème d'optimisation non-linéaire, la différence de coût est conséquente et se reflète sur la durée de reconstruction. Nous nous attachons donc à réduire le coût lié aux projections pour rendre la résolution plus rapide. La combinaison des projections orthogonales et de la mise à l'échelle constitue en quelque sorte le fil conducteur de ce mémoire.

1.3 Objectifs de recherche

Le but du projet est de développer et d'évaluer des méthodes d'optimisation capables de résoudre efficacement le problème de reconstruction (1.2) en coordonnées cylindriques.

Les résultats de McLaughlin (2017) sont encourageants, et pour cette raisons nous poursuivons dans la même direction que lui. Nous utilisons donc des méthodes à base de projections.

Nous nous concentrons sur les méthodes de Newton et de quasi-Newton, car elles sont utilisées et recommandées dans les travaux précédents. Nous disposons déjà d'une version en Matlab de l'algorithme TRON. Nous utilisons donc cette méthode, afin de pouvoir comparer nos

résultats à ceux de McLaughlin. Nous utiliserons également l'algorithme L-BFGS-B, car il est recommandé dans des travaux antérieurs (Hamelin, Goussard, et Dussault, 2010a).

Ce travail se place dans la continuité de ce qu'a fait McLaughlin. En particulier, nous nous attaquons au coût des projections orthogonales. Nous cherchons à développer des méthodes qui ne soient pas pénalisées par le calcul des projections tout en gardant l'accélération de convergence rendue possible par l'utilisation de la matrice C.

1.4 Plan du mémoire

Le mémoire s'organise autour d'un article de journal. Celui-ci a été soumis au journal *Optimization & Engineering*.

La section 2 contient une revue de littérature. On y présente quelques méthodes d'optimisation courantes en imagerie, des techniques de mise à l'échelle ainsi que les deux algorithmes cités dans la section précédente.

Nous décrivons rapidement la démarche adoptée au cours du travail de recherche avant de présenter l'article en section 4.

Nous terminons par une discussion des résultats et des choix décrits dans l'article, ainsi qu'une conclusion qui replace ce travail dans un cadre plus large.

CHAPITRE 2 REVUE DE LITTÉRATURE

Dans cette partie nous présentons quelques méthodes de directions projetées pour traiter le problème (1.4). Nous présentons d'abord les méthodes du premier ordre, courantes en imagerie, puis nous évoquons les procédures de mise à l'échelle mises en œuvre pour accélérer leur convergence. Enfin nous présentons deux méthodes d'ordre supérieur utilisées dans l'article.

2.1 Méthodes de premier ordre

Les méthodes de premier ordre sont revenues au goût du jour avec l'apparition de problèmes d'optimisation de grande taille qu'il n'est pas nécessaire de résoudre avec une tolérance stricte, notamment en imagerie, en analyse de données et en apprentissage machine. Simples, peu gourmandes en mémoire et en calculs, les méthodes de premier ordre semblent toutes indiquées pour de telles applications.

2.1.1 Gradient projeté

De nombreux algorithmes d'optimisation avec contraintes se basent sur la méthode du gradient projeté (Bertsekas, 1976). Un pas de gradient projeté partant de l'itéré courant x_k s'écrit

$$x(\alpha) = \text{Proj}(x_k - \alpha g_k), \tag{2.1}$$

où $g_k = \nabla f(x_k)$ et $\alpha > 0$. Une itération de la méthode s'écrit $x_{k+1} = x(\alpha_k)$, où la longueur de pas α_k est généralement calculée par une recherche linéaire d'Armijo.

La méthode de gradient projeté fait rapidement diminuer la fonction objectif, mais la convergence devient très lente au bout de quelques itérations. Pour cette raison, la méthode du gradient projeté est à réserver aux problèmes que l'on veut résoudre avec une tolérance large.

2.1.2 Variantes du gradient projeté

De nombreuses méthodes ont été développées pour accélérer la convergence des méthodes de premier ordre. En voici quelques-unes utilisées en imagerie.

Gradient projeté spectral

Afin d'accélérer la méthode de descente de gradient, Barzilai et Borwein (1988) proposent une stratégie inspirée de la méthode de la sécante pour choisir la longueur de pas α_k. Avec les notations $s_k = x_k - x_{k-1}$ et $y_k = g_{k-1} - g_k$, la longueur de pas est choisie entre

$$\alpha_k = \frac{\langle s_k, y_k \rangle}{\langle y_k, y_k \rangle} \qquad \text{et} \qquad \alpha_k = \frac{\langle s_k, s_k \rangle}{\langle s_k, y_k \rangle}. \tag{2.2}$$

La méthode résultante est qualifiée de gradient spectral car on cherche à estimer la courbure de la fonction objectif entre les deux derniers itérés.

Birgin et Martínez (2002) utilisent cette stratégie dans leur variante du gradient projeté. Lors d'une itération, on calcule un point intermédiaire $\tilde{x}_k = x(\alpha_k)$ en utilisant (2.2), puis on calcule le nouvel itéré x_{k+1} au moyen d'une recherche linéaire non-monotone entre x_k et $\tilde{x}_k$. Finalement une itération est de la forme

$$\left\{ \begin{array}{rcl} \tilde{x}_k & = & \text{Proj}(x_k - \alpha_k g_k) \\ x_{k+1} & = & x_k + \lambda_k \left(\tilde{x}_k - x_k \right), \end{array} \right. \tag{2.3}$$

où λ_k est déterminé au moyen d'une recherche linéaire non-monotone. On trouvera plus de détails et de références sur le gradient projeté spectral dans le mémoire de McLaughlin (2017).

Gradient accéléré

Nesterov (1983) propose une descente de gradient accélérée atteignant le taux de convergence $O(1/n^2)$, ce qui est optimal pour une descente de gradient. Dans cette version, un pas est une combinaison linéaire entre le pas précédent et un pas dans la direction de plus forte pente. On donne ainsi de l'inertie à la trajectoire des itérés. Cette inertie joue un rôle de régularisation pour la trajectoire et limite les zig-zags quand les itérés sont proches de la solution. Su, Boyd, et Candes (2014) étudient le gradient accéléré à l'aide d'équations différentielles ordinaires. Leur article contient par ailleurs de nombreuses références sur l'accélération des méthodes du premier ordre.

Beck et Teboulle (2009) utilisent cette accélération dans le cadre de la méthode du gradient proximal et proposent la méthode Fista *(Fast Iterative Shrinkage-Thresholding Algorithm)* destinée à des applications en traitement d'image. Dans le cadre du gradient projeté, une itération de Fista est de la forme

$$\left\{ \begin{array}{rcl} x_{k+1} & = & \text{Proj}(y_k - \alpha \nabla f(y_k)) \\ y_{k+1} & = & x_{k+1} + \frac{k-1}{k+2} \left(x_{k+1} - x_k \right). \end{array} \right. \tag{2.4}$$

L'accélération de Nesterov est régulièrement utilisée en reconstruction d'image. Kim, Ramani, et Fessler (2014) appliquent l'accélération à l'algorithme OS-SQS (*Ordered Subsets-Separable Quadratic Surrogates,* Erdoğan et Fessler, 1999b). Xu, Yang, Tan, Sawatzky, et Anastasio (2016) proposent un algorithme similaire basé sur Fista et décrivent sa mise en œuvre dans un cadre parallèle. Choi, Wang, Zhu, Suh, Boyd, et Xing (2010) utilisent le gradient accéléré pour traiter des problèmes de reconstruction à données limitées.

Sous-ensembles ordonnés

La méthode de sous-ensembles ordonnés est utilisée pour calculer approximativement le gradient d'un terme de moindres carrés $q = \|Ax-b\|_V$ en n'utilisant à chaque évaluation qu'une partie des données disponibles. On regroupe les lignes de la matrice du système A en m ensembles, qu'on utilise alternativement. Avec les notations de (1.2), on écrit par blocs

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix} \qquad V = \begin{pmatrix} V_1 & & \\ & \ddots & \\ & & V_m \end{pmatrix} \tag{2.5}$$

et

$$\|Ax - b\|_V = \|A_1 x - b_1\|_{V_1} + \cdots + \|A_m x - b_m\|_{V_m}. \tag{2.6}$$

La mise en œuvre des sous-ensembles ordonnés passe par une définition des sous-ensembles permettant d'avoir $q(x) = \|Ax-b\|_V \approx M\|A_\ell x - b_\ell\|_{V_\ell}$ pour $1 \leqslant \ell \leqslant m$. On approxime alors le gradient du terme de moindres carrés par

$$\nabla q(x) = A^{\mathrm{T}} V(Ax - b) \approx m A_\ell^{\mathrm{T}} V_\ell (A_\ell x - b_\ell). \tag{2.7}$$

Avec cette approximation, l'évaluation ne nécessite qu'une partie de la matrice du système, ce qui réduit le volume de calcul nécessaire.

Hudson et Larkin (1994) proposent une variante à sous-ensembles ordonnés de l'algorithme d'espérance-maximisation (Shepp et Vardi, 1982) pour la reconstruction en tomographie par émission, et Erdoğan et Fessler (1999b) présentent un algorithme à sous-ensembles ordonnés pour la tomographie par transmission. Il n'y a pas de théorie de convergence pour les méthodes à sous-ensembles ordonnés, mais elles sont beaucoup utilisées en tomographie. Kim *et al.* (2014) utilisent une stratégie de relaxation pour assurer la convergence de la méthode quand l'accélération de Nesterov est utilisée. Thibaudeau *et al.* (2013) discutent du choix des sous-ensembles dans le cas des coordonnées cylindriques.

2.1.3 Séparation du problème

Nous évoquons ici brièvement les approches par séparation du problème, car elles sont beaucoup utilisées en imagerie. Nous ne détaillons pas les algorithmes ni les hypothèses de régularité sur la fonction objectif et laissons le lecteur consulter les références proposées.

Comme de nombreux problèmes inverses, le problème de reconstruction est de la forme

$$\min_{x \in \mathbb{R}^n} \quad g(Qx) + h(x). \tag{2.8}$$

où le premier terme correspond à l'adéquation avec les données, et le deuxième terme est associé à la régularisation et aux contraintes. Dans notre cas, on a

$$g(z) = \tfrac{1}{2}\|z - V^{1/2}b\|_2^2 \qquad Q = V^{1/2}A \qquad h(x) = R(x) + \chi_{\mathbb{R}_n^+}(x), \tag{2.9}$$

où $\chi_{\mathbb{R}_n^+}(x)$ vaut zéro si $x \in \mathbb{R}_+^n$ et $+\infty$ sinon.

Il est possible de reformuler le problème (2.8) comme le problème

$$\min_{z=Qx} \quad g(z) + h(x), \tag{2.10}$$

et de le traiter avec des méthodes pour contraintes d'égalité (voir Nocedal et Wright, 2006, chapitre 17).

Une méthode populaire pour traiter un tel problème est le lagrangien augmenté (Hestenes, 1969). Xu (2017) présente des méthodes de premier ordre accélérées pour traiter les problèmes avec contraintes d'égalité. Nien et Fessler (2015) traitent le problème (2.10) à l'aide d'une variante linéarisée du lagrangien augmenté utilisant les sous-ensembles ordonnés.

Une autre possibilité est de traiter un problème de point-selle équivalent à (2.8), par exemple

$$\min_x \max_y \quad g(Qx) + \langle x, y \rangle - h^*(y), \tag{2.11}$$

où h^* est la fonction conjuguée de h au sens de Fenchel (voir Parikh et Boyd, 2013).

Sidky, Jørgensen, et Pan (2013) reconstruisent des images tomographiques à l'aide de la méthode primale-duale de Chambolle et Pock (2011). En imagerie, Labouesse, Allain, Idier, Bourguignon, Negash, Liu, et Sentenac (2017) utilisent l'algorithme de Condat (2013) pour traiter un problème de microscopie, sans effectuer de projections orthogonales.

L'ouvrage de Beck (2017) présente plus de méthodes pour problèmes séparés. La séparation du problème est intéressante quand la nature des contraintes rend les projections orthogonales

difficiles. L'approche par séparation est également utilisée dans un contexte proximal (voir Parikh et Boyd, 2013).

Dans ce mémoire, nous utilisons une pénalisation différentiable et les contraintes sont simples à traiter. Pour cette raison nous ne cherchons pas à séparer le problème, et nous n'en dirons pas plus sur cette approche.

2.2 Mise à l'échelle

Quand le problème (1.4) est mal conditionné, on peut accélérer la convergence des algorithmes au moyen d'une mise à l'échelle. Quand on met un problème à l'échelle, on change la manière dont on mesure les angles et les longueurs dans $\mathbb{R}^n$. La mise à l'échelle est souvent prise en compte dans la théorie de convergence des algorithmes génériques. Moré (1983) souligne l'importance de proposer des méthodes dont la convergence n'est pas remise en cause par une mise à l'échelle.

2.2.1 Changement de métrique

La mise à l'échelle consiste à choisir un produit scalaire autre que le produit scalaire euclidien $\langle \cdot, \cdot \rangle$ sur $\mathbb{R}^n$. Ce produit scalaire est utilisé en particulier pour calculer le gradient de la fonction objectif et les projections orthogonales.

On considère un produit scalaire de la forme $\langle \cdot, \cdot \rangle_{D^{-1}} = \langle \cdot, D^{-1} \cdot \rangle$, où D est une matrice symétrique définie positive. Afin de limiter le coût supplémentaire lié à la mise à l'échelle, on choisit de préférence une matrice D que l'on peut appliquer ou inverser facilement.

En développant f au premier ordre autour de x_k, on obtient

$$f(x_k + z) - f(x_k) = \langle g_k, z \rangle + \mathrm{o}(\|z\|) = \langle Dg_k, z \rangle_{D^{-1}} + \mathrm{o}(\|z\|). \tag{2.12}$$

Le gradient de f pour le nouveau produit scalaire est donc Dg_k. Pour cette raison, on cherche D comme un préconditionneur pour la hessienne $\nabla^2 f(x_k)$. En effet, si $D \approx \nabla^2 f(x_k)^{-1}$, la direction de descente Dg_k est proche de la direction de Newton.

Dans le cadre des contraintes de positivité, la projection orthogonale sur $\mathbb{R}^n_+$ est définie par

$$\mathrm{Proj}_D(x) = \arg\min_{v \geqslant 0} \|v - x\|_{D^{-1}}. \tag{2.13}$$

Il est intéressant de choisir D de manière à pouvoir calculer les projections par une formule explicite, ou au moins à coût limité.

Le choix du produit scalaire est un compromis à réaliser entre la qualité des directions de descente $-Dg_k$ engendrées, la compatibilité avec les contraintes et le coût du produit par D ou son inverse.

2.2.2 Mise à l'échelle diagonale

Les problèmes à contraintes de bornes sont souvent mis à l'échelle en utilisant une matrice D diagonale. Ce choix possède deux avantages. D'une part, la direction de descente $-Dg$ est facile à calculer de manière parallèle. D'autre part, on calcule les projections orthogonales en utilisant la formule directe (1.10).

Dans le cadre des méthodes à région de confiance, Conn, Gould, et Toint (1988) recommandent l'usage d'une mise à l'échelle diagonale pour les problèmes à contraintes de bornes.

Bonettini, Zanella, et Zanni (2008) utilisent une méthode de gradient projeté avec une mise à l'échelle diagonale dans un problème de débruitage d'image comprenant des contraintes de bornes et une contrainte de type $a^{\mathrm{T}}x \leqslant c$. Dans ce cas, la mise à l'échelle augmente peu le prix de la projection orthogonale. Les auteurs présentent plusieurs choix pour la matrice D_k, et mentionnent plusieurs stratégies de type Barzilai et Borwein (1988) pour la longueur de pas.

En reconstruction d'image, Erdoğan et Fessler (1999b) élaborent leur mise à l'échelle au moyen d'une approximation majorante de la fonction objectif de type

$$m_k(x_k + z) = f(x_k) + \langle z, g_k\rangle + \tfrac{1}{2}\langle z, D_k^{-1} z\rangle, \tag{2.14}$$

où D_k est diagonale et choisie telle que $m_k(x_k + z) \geqslant f(x_k + z)$ quand $x_k + z$ est réalisable. La direction de descente qui minimise ce modèle est $z = -D_k g_k$. On retrouve cette mise à l'échelle dans d'autres travaux du même laboratoire (Kim *et al.*, 2014; Nien et Fessler, 2015).

Dans leur algorithme primal-dual (Chambolle et Pock, 2011), Pock et Chambolle (2011) proposent également une mise à l'échelle diagonale portant sur la variable primale et la variable duale afin d'améliorer la convergence.

Parfois, le conditionnement du problème est trop mauvais pour obtenir une convergence rapide par une mise à l'échelle diagonale. La prochaine section décrit une astuce permettant d'appliquer un préconditionnement non diagonal sans changer les projections dans le cadre des contraintes de bornes.

2.2.3 Mise à l'échelle partiellement diagonale

Dans son article sur la méthode de Newton projetée, Bertsekas (1982) donne des résultats sur la combinaison des directions mises à l'échelle et les contraintes de bornes.

On cherche à faire décroître la fonction objectif par un pas de la forme

$$x(\alpha) = \mathrm{Proj}(x_k - \alpha D g_k), \tag{2.15}$$

où D est symétrique définie positive. Ici, deux métriques cohabitent : seul le gradient est mis à l'échelle, tandis que la projection est toujours calculée par la formule (1.10).

Bertsekas cherche des matrices D pour lesquelles la direction mise à l'échelle projetée $\dot{x}(0)$ (voir 1.1.3) est une direction de descente. Autrement dit, la direction projetée doit vérifier

$$\langle \dot{x}(0), g_k \rangle < 0. \tag{2.16}$$

On assure ainsi l'existence de pas α qui font décroître la fonction objectif. Cette condition n'est pas forcément vérifiée pour D non diagonale.

Bertsekas utilise une matrice D partiellement diagonale qui dépend des contraintes actives au point x_k. Si on réordonne les composantes de x_k en regroupant les indices qui appartiennent à l'ensemble actif $I_+(x_k)$ à la fin, on peut écrire

$$x_k = \begin{pmatrix} x_F \\ x_\perp \end{pmatrix} \quad \text{et} \quad g_k = \begin{pmatrix} g_F \\ g_\perp \end{pmatrix}, \tag{2.17}$$

où $x_F = (x_i)_{i \notin I_+(x_k)}$ est la composante de x_k selon la face active et $x_\perp = (x_i)_{i \in I_+(x_k)}$ est la composante orthogonale (au sens du produit scalaire usuel sur $\mathbb{R}^n$) à la face active. D'après la définition (1.8), on a donc $x_\perp = 0$ et $g_\perp > 0$.

Si $I_+(x) = \{r+1, \ldots, n\}$, on utilise une matrice de la forme

$$D = \left(\begin{array}{c|ccc} \bar{D} & & & \\ \hline & d^{r+1} & & \\ & & \ddots & \\ & & & d^n \end{array}\right), \tag{2.18}$$

où $\bar{D}$ est définie positive et les $d^{r+1}, \ldots, d^n$ sont strictement positifs. On dit que D est diagonale par rapport à $I_+(x)$.

Dans la structure (2.18), on a restreint la mise à l'échelle à la face active, alors qu'on ne

change pas les coordonnées de $-g_\perp$, qui sont mises à zéro par la projection (1.10). Le pas (2.15) et la direction $\dot{x}(0)$ s'écrivent respectivement

$$x(\alpha) = \begin{pmatrix} \max\left(x_F - \alpha \bar{D} g_F, 0\right) \\ x_\perp \end{pmatrix} \quad \text{et} \quad \dot{x}(0) = \begin{pmatrix} -\bar{D} g_F \\ 0 \end{pmatrix} \tag{2.19}$$

Bertsekas (1982, proposition 1) montre que une telle matrice D vérifie la condition (2.16). En effet, si x_k n'est pas un point critique, on a $g_F \neq 0$ et

$$\langle g_k, \dot{x}(0) \rangle = \left\langle g_F, -\bar{D} g_F \right\rangle < 0. \tag{2.20}$$

Dans le cadre de la méthode de Newton projetée, $\bar{D}$ est l'inverse d'une sous-matrice principale de la hessienne, soit $\bar{D}^{-1} = \left[\partial_{ij} f(x)\right]_{i,j \notin I_+(x)}$. Kim, Sra, et Dhillon (2010) utilisent le même principe en remplaçant la hessienne par une matrice de quasi-Newton.

Bonettini, Landi, Piccolomini, et Zanni (2013) présentent des méthodes faisant intervenir des matrices de mise à l'échelle diagonales par rapport à l'ensemble actif. Parmi ces méthodes, on retrouve une variante de la méthode de Newton projetée proposée par Landi et Loli Piccolomini (2008).

Gafni et Bertsekas (1984) présentent une autre manière de voir la mise à l'échelle partiellement diagonale, dans le cadre de contraintes générales. Dans leur approche, on applique la mise à l'échelle uniquement à la composante de $-g_k$ selon le cône des directions admissibles.

2.3 Méthodes d'ordre supérieur

Malgré les améliorations possibles, les méthodes de premier ordre progressent lentement après quelques itérations. Pour résoudre le problème avec une tolérance stricte, on préfèrera utiliser des méthodes d'ordre supérieur, qui possèdent en général une convergence superlinéaire.

La méthode de Newton projetée (Bertsekas, 1982), vue plus haut, est un exemple de méthode d'ordre 2, car elle fait intervenir les dérivées secondes de la fonction objectif. Les méthodes de quasi-Newton sont aussi considérées comme des méthodes d'ordre supérieur. En effet, dans les méthodes de quasi-Newton classiques, la différence entre la pseudo hessienne et la vraie hessienne tend vers zéro, ce qui permet d'atteindre un ordre de convergence superlinéaire (Dennis et Moré, 1977).

Nous présentons la forme générale d'une méthode de directions projetées, ainsi que la notion de point de Cauchy. Nous évoquons ensuite rapidement les algorithmes présentés dans l'article.

2.3.1 Structure générale

Dans les méthodes présentées, on représente la fonction objectif autour de l'itéré courant x_k par le modèle

$$m_k(x_k + z) = f(x_k) + \langle g_k, z\rangle + \tfrac{1}{2}\langle z, B_k z\rangle, \tag{2.21}$$

où B_k est la hessienne $\nabla^2 f(x_k)$ ou une matrice de quasi-Newton.

Une itération débute par l'identification d'un ensemble actif $\mathcal{A}_k$ et d'une face active $F_+(\mathcal{A}_k)$ dans laquelle on travaille. Dans les méthodes vues précédemment (Bertsekas, 1982; Kim *et al.*, 2010), on identifie l'ensemble actif directement à partir de x_k, soit $\mathcal{A}_k = I_+(x_k)$. Une possibilité plus adaptée aux grands problèmes est d'utiliser un point de Cauchy. On y revient à la section 2.3.2.

Pour calculer une direction, on résout ensuite (souvent approximativement) le problème restreint à la face active

$$\min_{x \in F_+(x_k)} \quad m_k(x). \tag{2.22}$$

La solution de (2.22) sert de base à la création d'un nouvel itéré x_{k+1}. L'itération se termine par le calcul du nouvel itéré suivi de mises à jour diverses.

La structure d'une telle méthode est résumée dans l'algorithme 2.1. Au bout d'un certain nombre d'itérations, l'ensemble actif $\mathcal{A}_k$ finit par coïncider avec $I_+(x^*)$ (Burke et Moré, 1994). On tient évidemment à trouver le bon ensemble actif le plus vite possible. Cela peut être facilité par l'usage d'un point de Cauchy.

Algorithme 2.1: Structure générale d'une méthode de directions projetées.

Initialiser $x_0 = 0$
tant que Conditions de convergence non respectées **faire**
  Identifier un ensemble actif $\mathcal{A}_k$ et une face active $F(\mathcal{A}_k)$
  Résoudre le sous-problème quadratique (2.22)
  Calculer le nouvel itéré x_{k+1}
  Effectuer les mises à jour nécessaires
fin

2.3.2 Point de Cauchy

Afin d'identifier rapidement l'ensemble actif final $I_+(x^*)$ pour les problèmes en grande dimension, on doit pouvoir ajouter ou enlever beaucoup de contraintes à la fois dans l'ensemble actif (Moré et Toraldo, 1991). Une manière de faciliter les variations de l'ensemble actif est d'utiliser un point de Cauchy x_k^{C}, calculé au moyen d'un pas de gradient projeté.

Le point de Cauchy est calculé par un pas de gradient projeté de la forme (2.1), en cherchant le premier minimum local du modèle (2.21) le long du chemin $x \mapsto x(\alpha)$. L'ensemble actif retenu pour la suite de l'itération est

$$\mathcal{A}^{\mathrm{C}} = \left\{ i \mid x_i^{\mathrm{C}} = 0 \text{ et } [\nabla f(x)]_i > 0 \right\}. \tag{2.23}$$

La recherche du point de Cauchy est illustrée en figure 2.1.

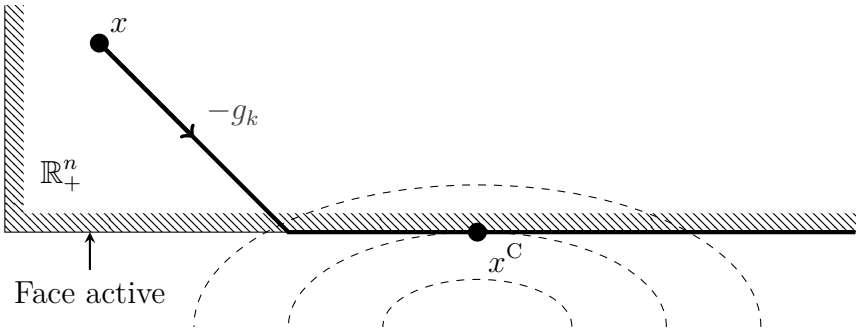

Figure 2.1 Le point de Cauchy minimise le modèle quadratique, dont on a dessiné les lignes de niveau en pointillés, le long du chemin projeté. Il est utilisé pour déterminer la face active.

Introduit dans le cadre des méthodes à région de confiance (Moré, 1983), le point de Cauchy assure souvent la convergence globale des algorithmes, tandis que la direction de descente basée sur le modèle quadratique détermine la vitesse de convergence. Conn *et al.* (1988) adaptent le point de Cauchy aux problèmes bornés. Dans ce cas, le point de Cauchy joue en plus le rôle de faciliter l'ajout de contraintes à l'ensemble actif. Mais la notion de point de Cauchy peut s'appliquer pour toutes sortes de contraintes, et Conn, Gould, Sartenaer, et Toint (1993) présentent un point de Cauchy généralisé, adapté à des contraintes très générales. Un tel point de Cauchy respecte des conditions relaxées et peut être issu par exemple de projections inexactes.

2.3.3 L-BFGS-B et TRON

Les algorithmes L-BFGS-B et TRON sont deux méthodes pour problèmes de grande taille à contraintes de bornes. On les présente en détail dans l'article.

L'algorithme L-BFGS-B (*Limited-memory BFGS for Bound contraints,* Byrd, Lu, Nocedal, et Zhu, 1995) est une méthode de quasi-Newton à mémoire limitée qui a déjà été utilisée en tomographie par émission (Kaplan, Haynor, et Vija, 1999) et par transmission (Hamelin *et al.*, 2010a). On y exploite massivement la structure de la formule L-BFGS compacte (Byrd, Nocedal, et Schnabel, 1994) pour effectuer des opérations très efficacement dans le contexte

des contraintes de bornes. En particulier, la recherche du point de Cauchy se fait de manière exacte en examinant successivement chaque segment du chemin projeté (2.1). De même, on résout (2.22) en résolvant directement un système linéaire par la formule de Sherman-Morrison-Woodbury. Ainsi, l'algorithme L-BFGS-B est difficilement dissociable de la formule L-BFGS compacte et de ces procédures auxquelles il doit son efficacité.

L'algorithme TRON (*Trust-Region Newton method,* Lin et Moré, 1999) est une méthode de Newton à région de confiance. Ici, le modèle (2.21) peut être cher à évaluer car il se base sur la hessienne $\nabla^2 f(x_k)$. Pour cette raison, on utilise plutôt des procédures itératives. Ainsi, le point de Cauchy est calculé de manière inexacte par une méthode de rebroussement et vérifie une condition de type Armijo afin d'assurer la convergence de la méthode. La résolution du sous-problème (2.22), elle, fait intervenir la méthode du gradient conjugué.

Dans l'algorithme TRON, l'accent est mis sur la plasticité de l'ensemble actif. Celui-ci, grâce à de nombreuses recherches projetées, peut varier facilement au cours d'une itération.

On trouvera plus de détails sur ces algorithmes à la section 4.4.

CHAPITRE 3 DÉMARCHE DE L'ENSEMBLE DU TRAVAIL

Les méthodes développées durant la maîtrise ainsi que les résultats obtenus sont décrits dans l'article.

Nous appliquons les méthodes de directions projetées au problème de reconstruction (1.2), dont la principale difficulté est la gestion de la mise à l'échelle. McLaughlin (2017) a opté pour un changement de variable, soulevant une autre difficulté liée au calcul des projections orthogonales. Nous cherchons à éviter cet écueil en travaillant avec la variable originale et en utilisant des directions de descente mises à l'échelle. On traite ainsi un problème à contraintes de bornes, pour lequel une méthode de directions projetées est toute indiquée.

L'article décrit la stratégie de mise à l'échelle utilisée. Celle-ci s'inspire directement du changement de variable. Ainsi, l'intégration de la mise à l'échelle dans les différentes procédures des algorithmes cherche à reproduire les effets du changement de variable. Cependant, on impose de pouvoir calculer les projections orthogonales à l'aide de la formule explicite (1.10). Pour cette raison, la mise à l'échelle des directions fait intervenir les opérateurs partiellement diagonaux vus au chapitre précédent, créés à partir de C. On présente dans l'article l'application de la mise à l'échelle sur les pas de gradient projeté, la méthode du gradient conjugué et les matrices de quasi-Newton, ainsi que son intégration dans les algorithmes TRON et L-BFGS-B présentés au chapitre précédent.

Nous montrons dans l'article des résultats numériques obtenus à l'aide de nos méthodes mises à l'échelle. Les problèmes considérés sont deux problèmes de reconstruction d'image en dimension 2 à partir de données synthétiques. Pour ces deux problèmes, la vitesse de convergence est comparée entre les méthodes avec mise à l'échelle et celles sans mise à l'échelle. De plus, dans le cas de TRON, on compare ces deux méthodes à l'approche avec changement de variable de McLaughlin (2017). On rassemble également des statistiques d'exécution pour savoir si les procédures des différents algorithmes représentent la même part du temps de calcul final. On propose aussi un essai avec le gradient projeté sprectral mis à l'échelle, et ce afin de justifier l'usage de méthodes d'ordre supérieur.

CHAPITRE 4 ARTICLE 1 : SCALED PROJECTED-DIRECTIONS METHODS WITH APPLICATION TO TRANSMISSION TOMOGRAPHY

Scaled Projected-Directions Methods
with Application to Transmission Tomography

Guillaume Mestdagh, Yves Goussard and Dominique Orban

Manuscript submitted to *Optimization and Engineering*

Abstract Statistical image reconstruction in X-Ray computed tomography yields large-scale regularized linear least-squares problems with nonnegativity bounds, where the memory footprint of the operator is a concern. Discretizing images in cylindrical coordinates results in significant memory savings but deteriorates the conditioning of the operator. We improve the Hessian conditioning by way of a block-circulant scaling operator and we propose a strategy to handle nondiagonal scaling in the context of projected-directions methods for bound-constrained problems. We describe our implementation of the scaling strategy using two algorithms: TRON, a trust-region method with exact second derivatives, and L-BFGS-B, a linesearch method with a limited-memory quasi-Newton Hessian approximation. We compare our approach with one where a change of variable is made in the problem. On two reconstruction problems, our approach converges faster than the change of variable approach, and achieves much tighter accuracy than a first-order method.

Keywords X-Ray CT Reconstruction Projected-Directions Methods Scaling

4.1 Introduction

We consider the bound-constrained problem

$$\min \; f(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \geqslant 0, \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and $\mathcal{C}^2$. We assume that $\nabla^2 f$ cannot be stored explicitly or in factorized form. We are particularly interested in the case where (4.1) is large and badly scaled. Our main motivation is to solve efficiently statistical image reconstruction problems arising from X-Ray Computed Tomography (CT) (Herman, 2009). Whereas cartesian coordinates are typical, discretizing such problems in cylindrical coordinates yields large savings in storage, but results in badly scaled problems and, without proper scaling, off-the-shelf

solvers usually fail.

In this paper we employ a scaling strategy that exploits the structure of f combined with the trust-region projected Newton method of Lin et Moré (1999) and with the line search limited-memory BFGS for bound-constrained problems of Byrd *et al.* (1995) to maintain satisfaction of the bound constraints at all times.

Motivation and previous work

Our main interest resides in statistical image reconstruction problems arising from X-Ray Computed Tomography (CT) (Herman, 2009). Compared to analytical methods such of the filtered backprojection family (Feldkamp *et al.*, 1984), statistical reconstruction results in less noisy images but is more computationally expensive (Fessler, 2000).

Sauer et Bouman (1993) show that an image $\mathbf{x}$ can be estimated from the measurements $\mathbf{b}$ by solving

$$\min \ \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathbf{V}}^2 + \lambda R(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \geqslant 0, \tag{4.2}$$

where $\mathbf{A}$ is a large sparse matrix, $\lambda > 0$ is a regularization parameter, $R : \mathbb{R}^n \to \mathbb{R}$ is a convex regularization function and $\mathbf{V}$ is a diagonal weight matrix with $V_{ii} = \exp(-b_i)$. In (4.2), the objective is composed of a least-squares data-fitting term, and a regularization term to discourage large differences between adjacent pixels. Here, we focus on situations where one wishes to solve problem (4.2) precisely (with a low tolerance level), which corresponds to instances where the information present in the data should be fully exploited, as, e.g., in low-dose CT reconstruction or for imaging thin structures in micro-CT (Hamelin, Goussard, Dussault, Cloutier, Beaudoin, et Soulez, 2010b).

While reconstructed images are usually discretized using a cartesian grid of voxels (Fessler, 2000, equation (6)), we use cylindrical coordinates (Thibaudeau *et al.*, 2013). This discretization is in adequation with the circular geometry of the data acquisition process and results in block-circulant $\mathbf{A}$, which has far lower storage requirements than the operator resulting from cartesian coordinates (Goussard *et al.*, 2013).

In statistical image reconstruction, first-order methods are often preferred due to their low storage and computational demands, and their simplicity (Fessler, 2000). Usual reconstruction methods include the expectation-maximization algorithm (Lange et Fessler, 1995; Ahn, Fessler, Blatt, et Hero, 2006), coordinate-descent methods (Sauer et Bouman, 1993; Noo, Hahn, Schöndube, et Stierstorfer, 2016) and gradient-based methods (Erdoğan et Fessler, 1999a; Kim *et al.*, 2014). This last category is probably the most studied in image reconstruction. Such methods are often improved by using ordered subsets (Erdoğan et Fessler,

1999b; Hudson et Larkin, 1994) or Nesterov momentum (Nesterov, 1983; Kim *et al.*, 2014; Xu *et al.*, 2016; Choi *et al.*, 2010; Jensen, Jørgensen, Hansen, et Jensen, 2012). Recently, problem-splitting and proximal approaches have been proposed in the context of sparse reconstruction (Sidky *et al.*, 2013; Nien et Fessler, 2015; Xu *et al.*, 2016).

For imaging applications, diagonal scaling is usually sufficient for first-order methods to perform well (Pock et Chambolle, 2011; Bonettini *et al.*, 2008). In CT reconstruction, diagonal scaling is sometimes referred to as the *separable quadratic surrogate method* (Erdoğan et Fessler, 1999b). However, in cylindrical coordinates, widely different voxel sizes make $\mathbf{A}$ badly scaled, and diagonal scaling is no longer appropriate.

To improve the conditioning of (4.2), we follow Golkar (2013) and use a block-circulant scaling operator $\mathbf{C}$ that exploits the block-circulant property of $\mathbf{A}$ and of the finite-difference matrices that appear in the regularization term $R(\mathbf{x})$. The scaling operator can be written

$$\mathbf{C} = \mathbf{F}^\star \mathbf{T} \mathbf{F}, \tag{4.3}$$

where $\mathbf{T}$ is diagonal, $\mathbf{F}$ is a discrete block Fourier transform, and a star indicates the conjugate transpose. Thus, $\mathbf{C}$ and its inverse can be applied to a vector at the cost of a fast Fourier transform, namely in $\mathrm{O}(s_b n_b \log n_b)$ operations, where s_b is the size of a square block, and n_b is the number of blocks.

If $\mathbf{C}$ satisfies $\mathbf{C}^{\mathrm{T}} \nabla^2 f(\mathbf{x}) \mathbf{C} \approx \mathbf{I}$, the change of variable $\mathbf{x} = \mathbf{C}\mathbf{u}$ transforms (4.1) into the scaled problem

$$\min \ \tfrac{1}{2} \|\mathbf{A}\mathbf{C}\mathbf{u} - \mathbf{b}\|_{\mathbf{V}}^2 + \lambda R(\mathbf{C}\mathbf{u}) \quad \text{s.t. } \mathbf{C}\mathbf{u} \geqslant 0, \tag{4.4}$$

in which the objective Hessian is better conditioned than in (4.2). However, (4.4) features linear inequality constraints instead of simple bounds.

McLaughlin (2017) solves (4.4) with `Cflash`, a variant of TRON (Lin et Moré, 1999) adapted to linear inequality constraints. Each iteration of `Cflash` requires projecting candidate iterates $\mathbf{u}$ into the feasible set by solving

$$\min_{\mathbf{v}} \ \|\mathbf{v} - \mathbf{u}\| \quad \text{s.t. } \mathbf{C}\mathbf{v} \geqslant 0, \tag{4.5}$$

which represents a significant amount of computation. In `Cflash`, the above projection is computed efficiently by solving the dual problem, which is a bound-constrained linear least-squares problem with operator $\mathbf{C}$, iteratively. Even though (4.5) can be solved efficiently thanks to the structure of $\mathbf{C}$, it remains substantially more costly than projecting into simple bounds.

Instead of solving (4.4), we propose to solve (4.2) with a scaled quasi-Newton and Newton method in order to reproduce the effect of a change of variable without actually performing it. An advantage of scaled directions is that we can choose the scaling at each iteration.

Bonettini *et al.* (2008) describe diagonally-scaled projection methods in which both the gradient and projection subproblems are scaled. Bonettini *et al.* (2013) extend the approach to block-diagonal scaling in the context of image deblurring, a problem where the system operator is a 2D convolution, which is block-circulant with circulant blocks. Their scaling depends on the active constraints at the current iterate, as in the projected Newton method described by Bertsekas (1982), which allows them to apply a nondiagonal scaling while preserving simple projections. They also investigate a quasi-Newton method where the Hessian approximation is a truncated spectral decomposition of the system matrix. In the partially-diagonal scaling approach, Bonettini *et al.* apply the method of Landi et Loli Piccolomini (2008) and solve the linear system inexactly using the conjugate gradient method (CG) of Hestenes et Stiefel (1952). Both methods are presented as scaled gradient projection methods in which the scaling operator is inspired from the problem Hessian or from a quasi-Newton matrix.

Our approach extends that of Bonettini *et al.* (2008) to more complex methods for bound-constrained problems. Because we apply the scaling to higher-order methods, we restrict our study to the situation where the scaling operator can be applied or inverted easily. The problem Hessian does not enter this category, because it is expensive to apply and we need to use CG to solve a linear system involving it. Unlike Bonettini *et al.*, we consider the use of partially-diagonal operators in the context of a change of scalar product, independently from the solution method. We then describe the impact of the scaling on families of general-purpose optimization algorithms. Specifically, we consider the limited-memory BFGS method for bound-constrained problems, and a trust-region projected Newton method. The convergence properties of both methods rest on the computation of a Cauchy point, i.e., an approximate minimizer in the negative gradient direction. Our implementation of L-BFGS-B differs from that of Zhu, Byrd, Lu, et Nocédal (1997) in that we compute an inexact Cauchy point and restrict the computation of a step to an active face by way of restriction operators. Our implementation of a projected Newton method follows the design of TRON (Lin et Moré, 1999), in which projected gradient steps are performed to identify an inexact Cauchy point and a candidate active set, followed by a sequence of Newton steps on the active face of the feasible set globalized by a trust-region mechanism. Such an approach has been shown to be able to add and remove numerous bounds from the active set at a time and to be particularly appropriate for large-scale problems. We illustrate the performance of both methods on synthetic images, and we compare it to that of the projected spectral gradient

method, a classic method in imaging.

Notation Lowercase and uppercase bold Latin letters represent vectors and matrices, respectively. Light face Latin letters represent integers, such as iteration counters, and functions. In addition, the i-th component of $\mathbf{x}$ is denoted x_i and the (i, j)-th element of $\mathbf{A}$ is A_{ij}. Light face Greek letters represent scalars. The Euclidean scalar product on $\mathbb{R}^n$ is denoted $\langle \cdot, \cdot \rangle$. The i-th partial derivative of function f at x is denoted $\partial_i f(x)$.

4.2 A scaling strategy for bound-constrained problems

In this section we describe the effect of scaling on procedures that are common between the two methods we present. After a short presentation of the scaling strategy, we present two procedures we use in L-BFGS-B and TRON to compute a Cauchy point and a descent step in the active face respectively. We then apply the scaling on the limited-memory quasi-Newton matrix that appears in L-BFGS-B. We integrate these procedures into the chosen algorithms in Section 4.3.

4.2.1 Overview of the strategy

Our scaling strategy consists in using a metric in which the problem is well scaled and the projections can be computed with a direct formula.

A linear change of variables is equivalent to changing the scalar product in the original space. Indeed, if $\mathbf{x} = \mathbf{C}\mathbf{u}$ and $\mathbf{z} = \mathbf{C}\mathbf{w}$,

$$\langle \mathbf{u}, \mathbf{w} \rangle = \left\langle \mathbf{x}, \mathbf{P}^{-1}\mathbf{z} \right\rangle, \qquad \mathbf{P} := \mathbf{C}\mathbf{C}^{\mathrm{T}}. \tag{4.6}$$

This equivalence makes it possible to import geometric elements from the scaled space into the original space. From now on, we use $\mathcal{X}$ to denote the original space and $\mathcal{U}$ for the scaled space. Every $\mathbf{x} \in \mathcal{X}$ corresponds to a $\mathbf{u} \in \mathcal{U}$ such that $\mathbf{x} = \mathbf{C}\mathbf{u}$. We denote $\bar{f}$ the objective function of (4.1) in the scaled space, i.e., for $\mathbf{u} \in \mathcal{U}$,

$$\bar{f}(\mathbf{u}) := f(\mathbf{C}\mathbf{u}), \qquad \nabla \bar{f}(\mathbf{u}) = \mathbf{C}^{\mathrm{T}} \nabla f(\mathbf{C}\mathbf{u}), \qquad \nabla^2 \bar{f} = \mathbf{C}^{\mathrm{T}} \nabla^2 f(\mathbf{C}\mathbf{u}) \mathbf{C}. \tag{4.7}$$

The first element we transform is the gradient direction. Indeed, due to the choice of scaling, the gradient direction in $\mathcal{U}$ is expected to be a more promising descent direction than the

gradient direction in $\mathcal{X}$. If $\mathbf{u} \in \mathcal{U}$, $\mathbf{x} = \mathbf{C}\mathbf{u}$, and $\alpha > 0$,

$$\mathbf{x}(\alpha) := \mathbf{C}(\mathbf{u} - \alpha \nabla \bar{f}(\mathbf{u})) = \mathbf{x} - \alpha \mathbf{C} \nabla \bar{f}(\mathbf{u}) = \mathbf{x} - \alpha \mathbf{P} \nabla f(\mathbf{x}). \tag{4.8}$$

In other words, a step along the negative gradient in $\mathcal{U}$ is equivalent to a step in the direction

$$\mathbf{q} = -\mathbf{P} \nabla f(\mathbf{x}) \tag{4.9}$$

in $\mathcal{X}$. We use (4.9) instead of $-\nabla f(\mathbf{x})$ in the hope that the scaled search direction better captures natural problem curvature.

4.2.2 Projected directions

In the context of projected methods, it is often necessary to work on a face of the feasible set, or to take projected gradient steps.

Projected gradient steps

Because the method of Lin et Moré (1999) is related to that of Bertsekas (1982), it is useful to review certain common basic concepts.

A standard projected gradient step from $\mathbf{x}$ can be described by $\text{Proj}\left(\mathbf{x} - \alpha \nabla f(\mathbf{x})\right) - \mathbf{x}$ where $\alpha > 0$. A scaled projected gradient step has the form $\mathbf{x}(\alpha) - \mathbf{x}$, where

$$\mathbf{x}(\alpha) = \text{Proj}\left(\mathbf{x} + \alpha \mathbf{d}\right), \tag{4.10}$$

and where $\mathbf{d}$ is a linear transformation of $\nabla f(\mathbf{x})$. The direction $\mathbf{d}$ must be a descent direction in the sense that there exists $\bar{\alpha} > 0$ such that $f(\mathbf{x}(\alpha)) < f(\mathbf{x})$ for all $\alpha \in (0, \bar{\alpha}]$.

Bertsekas (1982, section 2) explains that (4.9) might not be such a descent direction. We define the *binding* constraints at $\mathbf{x}$ as those with indices in

$$I_+(\mathbf{x}) = \{i \mid x_i = 0 \text{ and } \partial_i f(\mathbf{x}) > 0\}, \tag{4.11}$$

where $\partial_i f(\mathbf{x})$ is the i-th component of $\nabla f(\mathbf{x})$. We introduce the subspace

$$F = \{\mathbf{a} \in \mathbb{R}^n \mid a_i = 0 \text{ for all } i \in I_+(\mathbf{x})\}, \tag{4.12}$$

and the set $F_+ = F \cap \mathbb{R}^n_+$, called the face of the feasible set exposed by $-\nabla f(\mathbf{x})$. Bertsekas

(1982, Proposition 1) establishes that

$$\mathbf{d} = -\bar{\mathbf{P}}\nabla f(\mathbf{x}), \tag{4.13}$$

where the matrix $\bar{\mathbf{P}}$ is defined by

$$\bar{\mathbf{P}}_{ij} = \begin{cases} \mathbf{P}_{ij} & \text{if } i, j \notin I_+(\mathbf{x}) \\ 0 & \text{otherwise,} \end{cases} \tag{4.14}$$

is a descent direction in the sense defined above.

To compute (4.13), we decompose $\mathbb{R}^n = F \oplus G$ where $G = F^\perp$, and we write

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \mathbf{g}_F \\ \mathbf{g}_G \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} \mathbf{P}_{FF} & \mathbf{P}_{FG} \\ \mathbf{P}_{GF} & \mathbf{P}_{GG} \end{pmatrix}, \tag{4.15}$$

where $\mathbf{g}_F = (\partial_i f(\mathbf{x}))_{i \notin I_+(\mathbf{x})}$ and $\mathbf{g}_G = (\partial_i f(\mathbf{x}))_{i \in I_+(\mathbf{x})}$.

The gradient is first projected onto the subset F. Then the scaling is made on the projected gradient $\mathbf{g}_F$ by applying the principal submatrix $\mathbf{P}_{FF}$. This submatrix is obtained by keeping only the rows and columns whose indices are not in $I_+(\mathbf{x})$. Finally,

$$\mathbf{d} = -\begin{pmatrix} \mathbf{P}_{FF} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{g}_F \\ \mathbf{g}_G \end{pmatrix} = \begin{pmatrix} -\mathbf{P}_{FF}\,\mathbf{g}_F \\ 0 \end{pmatrix}. \tag{4.16}$$

Conjugate gradient in a face of the feasible set

The same procedure can be used to modify conjugate gradient directions inside a face of the feasible set. Consider the quadratic problem

$$\min_{x \in F} \tfrac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{B}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{g}, \tag{4.17}$$

To solve (4.17), the Conjugate Gradient method of Hestenes et Stiefel (1952) is applied to the equivalent reduced problem

$$\min_{\bar{x} \in \mathbb{R}^{\dim F}} \tfrac{1}{2}\bar{\mathbf{x}}^{\mathrm{T}}\mathbf{B}_{FF}\bar{\mathbf{x}} + \bar{\mathbf{x}}^{\mathrm{T}}\bar{\mathbf{g}}. \tag{4.18}$$

The directions $\bar{\mathbf{p}}_1, \bar{\mathbf{p}}_2, \ldots$ generated by the procedure are conjugate with respect to the principal submatrix $\mathbf{B}_{FF}$ of $\mathbf{B}$ (Hestenes et Stiefel, 1952). In particular, at the k-th iteration,

the next direction is defined as

$$\bar{\mathbf{p}}_{k+1} = \bar{\mathbf{r}}_{k+1} + \beta_k \bar{\mathbf{p}}_k, \tag{4.19}$$

where $\bar{\mathbf{r}}_{k+1} = \bar{\mathbf{g}} - \mathbf{B}_{FF}\bar{\mathbf{x}}_{k+1}$ is the residual and β_k is chosen so that the conjugacy condition $\bar{\mathbf{p}}_{k+1}^{\mathrm{T}}\mathbf{B}_{FF}\bar{\mathbf{p}}_k = 0$ is verified.

To improve the convergence of CG, we use a scaled residual to generate the new direction. The direction update formula becomes

$$\bar{\mathbf{p}}_{k+1} = \mathbf{P}_{FF}\bar{\mathbf{r}}_{k+1} + \beta'_k \bar{\mathbf{p}}_k, \tag{4.20}$$

where β'_k is chosen to respect the conjugacy condition. Note that applying the scaling in the case of CG, is equivalent to preconditioning CG with $\mathbf{P}_{FF}$.

4.2.3 Limited memory quasi-Newton matrices

In a quasi-Newton method, the objective function is approximated about the current iterate $\mathbf{x}_k$ by the quadratic model

$$m_k(\mathbf{x}_k + \mathbf{z}) = f_k + \mathbf{g}_k^{\mathrm{T}}\mathbf{z} + \tfrac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{B}_k\mathbf{z}, \tag{4.21}$$

where f_k and $\mathbf{g}_k$ are the objective value and gradient at $\mathbf{x}_k$ respectively, and $\mathbf{B}_k = \mathbf{B}_k^T$ is an approximation of $\nabla^2 f(\mathbf{x}_k)$. Secant methods are a special case in which $\mathbf{B}_k$ must satisfy the secant equation

$$\mathbf{B}_k\mathbf{s}_{k-1} = \mathbf{y}_{k-1}, \tag{4.22}$$

where $\mathbf{s}_{k-1} := \mathbf{x}_k - \mathbf{x}_{k-1}$ and $\mathbf{y}_{k-1} := \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$. Secant methods typically define $\mathbf{B}_k$ as rank-one or rank-two update of $\mathbf{B}_{k-1}$ involving $\mathbf{s}_{k-1}$ and $\mathbf{y}_{k-1}$. This has the disadvantage that $\mathbf{B}_k$ is almost always dense even though $\nabla^2 f(\mathbf{x}_k)$ might be sparse. Therefore, the entire matrix $\mathbf{B}_k$ must be stored, which is unrealistic in large-scale applications. However, at least conceptually, a product between $\mathbf{B}_k$ and a vector could be computed without storing $\mathbf{B}_k$ if the initial approximation $\mathbf{B}_0$ along with all the pairs $\{\mathbf{s}_i, \mathbf{y}_i\}_{0 \leq i \leq k-1}$ are stored instead.

In a limited-memory context, we store an initial matrix $\mathbf{B}_0$ along with the m most recent pairs $\{\mathbf{s}_i, \mathbf{y}_i\}_{k-m \leqslant i \leqslant k-1}$, where m is the memory. The procedure uses the information from $\mathbf{B}_0$ and from the m pairs to update and compute a product with $\mathbf{B}_k$. Even though $\mathbf{B}_k$ would still be dense if it were materialized, it it only represented implicitly.

Because the memory m is often small compared to the problem dimension, quasi-Newton updates can only contribute a limited amount of information to $\mathbf{B}_k$. For this reason, the

choice of $\mathbf{B}_0$ is critical to obtain a good approximation of the objective Hessian. In particular, when the Hessian is ill conditioned, choosing $\mathbf{B}_0$ as a multiple of the identity might lead to poor performance.

The best-know limited-memory quasi-Newton method is probably the limited-memory BFGS method (Nocedal, 1980), which additionally ensures that $\mathbf{B}_k$ is positive definite provided that $\mathbf{B}_{k-1}$ is positive definite and $\mathbf{s}_{k-1}^T\mathbf{y}_{k-1} > 0$. Several procedures exist to compute a product between $\mathbf{B}_k$ or its inverse and a vector, including the two-loop recursion (Nocedal, 1980), and a variant based on compact storage (Byrd *et al.*, 1994). We present the second one because it was designed to handle bound constraints.

The pairs $\{\mathbf{s}_i, \mathbf{y}_i\}_{k-m \leqslant i \leqslant k-1}$ are stored in two matrices

$$\mathbf{S} = \begin{pmatrix} \mathbf{s}_{k-m} & \cdots & \mathbf{s}_{k-1} \end{pmatrix} \quad \text{and} \quad \mathbf{Y} = \begin{pmatrix} \mathbf{y}_{k-m} & \cdots & \mathbf{y}_{k-1} \end{pmatrix}, \tag{4.23}$$

and we define

$$\mathbf{D} = \begin{pmatrix} \mathbf{s}_{k-m}^{\mathrm{T}}\mathbf{y}_{k-m} & & \\ & \ddots & \\ & & \mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1} \end{pmatrix} \quad \text{and} \quad \mathbf{L} = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ \mathbf{s}_{k-m+1}^{\mathrm{T}}\mathbf{y}_{k-m} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-m} & \cdots & \mathbf{s}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-2} & 0 \end{pmatrix}. \tag{4.24}$$

In the compact formula, $\mathbf{B}_k$ is stored implicitly as $\mathbf{B}_0$,

$$\mathbf{W} = \begin{pmatrix} \mathbf{Y} & \mathbf{B}_0\mathbf{S} \end{pmatrix} \quad \text{and} \quad \mathbf{M} = \begin{pmatrix} -\mathbf{D} & \mathbf{L}^{\mathrm{T}} \\ \mathbf{L} & \mathbf{S}^{\mathrm{T}}\mathbf{B}_0\mathbf{S} \end{pmatrix}^{-1}, \tag{4.25}$$

such that

$$\mathbf{B} = \mathbf{B}_0 - \mathbf{W}\mathbf{M}\mathbf{W}^{\mathrm{T}}. \tag{4.26}$$

where $\mathbf{B}_0$ is positive definite.

In most implementations, $\mathbf{B}_0$ is chosen as

$$\mathbf{B}_0 = \theta\mathbf{I} \quad \text{with} \quad \theta = \frac{\mathbf{y}_{k-1}^{\mathrm{T}}\mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^{\mathrm{T}}\mathbf{s}_{k-1}}, \tag{4.27}$$

where θ is a scaling parameter (Byrd *et al.*, 1995). A diagonal $\mathbf{B}_0$ leads to very efficient operations with the L-BFGS compact formula. However, it might be inappropriate for approximating ill-conditioned Hessians.

We choose $\mathbf{B}_0$ so that the L-BFGS operator in $\mathcal{X}$ reproduces the behavior of a standard

L-BFGS operator with initial approximation (4.27) in $\mathcal{U}$.

Assume that, in $\mathcal{U}$, $\bar{f}$ is approximated about the current scaled iterate $\mathbf{u}_k$ by the quadratic model

$$m'_k(\mathbf{u}_k + \mathbf{w}) = \bar{f}(\mathbf{u}_k) + \nabla \bar{f}(\mathbf{u}_k)^{\mathrm{T}}\mathbf{w} + \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{B}'_k\mathbf{w}, \tag{4.28}$$

where $\mathbf{B}'_k$ is a L-BFGS operator with initial approximation (4.27). The pairs $\{\bar{\mathbf{s}}_i, \bar{\mathbf{y}}_i\}$ in $\mathcal{U}$ are related to the pairs $\{\mathbf{s}_i, \mathbf{y}_i\}$ in $\mathcal{X}$ via

$$\bar{\mathbf{s}}_i = \mathbf{C}^{-1}\mathbf{s}_i \qquad \bar{\mathbf{y}}_i = \mathbf{C}^{\mathrm{T}}\mathbf{y}_i, \qquad i = k-m, \ldots, k-1. \tag{4.29}$$

We replace $\{\mathbf{s}_i, \mathbf{y}_i\}$ with $\{\bar{\mathbf{s}}_i, \bar{\mathbf{y}}_i\}$ in (4.26) and (4.27), and obtain

$$\mathbf{B}'_k = \bar{\theta}\mathbf{I} - \begin{pmatrix}\mathbf{C}^{\mathrm{T}}\mathbf{Y} & \bar{\theta}\mathbf{C}^{-1}\mathbf{S}\end{pmatrix} \begin{pmatrix}-\mathbf{D} & \mathbf{L}^{\mathrm{T}} \\ \mathbf{L} & \bar{\theta}\mathbf{S}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{S}\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{Y}^{\mathrm{T}}\mathbf{C} \\ \bar{\theta}\mathbf{S}\mathbf{C}^{-\mathrm{T}}\end{pmatrix}, \tag{4.30}$$

where

$$\bar{\theta} = \frac{\bar{\mathbf{y}}_{k-1}^{\mathrm{T}}\bar{\mathbf{y}}_{k-1}}{\bar{\mathbf{s}}_{k-1}^{\mathrm{T}}\bar{\mathbf{y}}_{k-1}} = \frac{\mathbf{y}_{k-1}^{\mathrm{T}}\mathbf{P}\mathbf{y}_{k-1}}{\mathbf{y}_{k-1}^{\mathrm{T}}\mathbf{s}_{k-1}}. \tag{4.31}$$

We use (4.21) to approximate f in $\mathcal{X}$. A comparison between (4.21) and (4.28) yields

$$\mathbf{B}'_k = \mathbf{C}^{\mathrm{T}}\mathbf{B}_k\mathbf{C}. \tag{4.32}$$

Finally $\mathbf{B}$ is a L-BFGS operator with initial approximation

$$\mathbf{B}_0 = \bar{\theta}\mathbf{C}^{-\mathrm{T}}\mathbf{C}^{-1} = \bar{\theta}\mathbf{P}^{-1} \tag{4.33}$$

Apart from the storage of $\mathbf{P}$, this modification does not require more storage in the compact L-BFGS formula. Instead of storing $\mathbf{S}$ and $\mathbf{S}^{\mathrm{T}}\mathbf{S}$, we store $\mathbf{P}^{-1}\mathbf{S}$ and $\mathbf{S}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{S}$, while $\mathbf{L}$ and $\mathbf{D}$ remain unchanged. The L-BFGS update only requires one product with $\mathbf{P}^{-1}$ to compute $\mathbf{P}^{-1}\mathbf{s}_k$ and one scalar product defined by $\mathbf{P}$ to compute $\bar{\theta}$.

4.3 Modified algorithms

In this section, we present the salient elements of the L-BFGS and TRON algorithms, and of our implementations. Then we describe the modification we brought to apply the scaling strategy.

4.3.1 The L-BFGS-B algorithm

The L-BFGS-B algorithm of Byrd *et al.* (1995) is a popular quasi-Newton method for bound-constrained problems. Its standard implementation exploits the compact representation of limited-memory quasi-Newton operators, a diagonal $\mathbf{B}_0$, and the Sherman-Morrison-Woodbury formula to solve linear systems whose coefficient is a principal submatrix corresponding to inactive indices. In our application, $\mathbf{B}_0$ is nondiagonal and its principal submatrices are not structured, so the Sherman-Morrison-Woodbury approach would be inefficient.

Presentation of the algorithm

At the beginning of an iteration, we compute the Cauchy point $\mathbf{x}_k^{\mathrm{C}}$ as the *exact* first local minimizer of (4.21) along the piecewise affine path

$$t \mapsto \mathrm{Proj}(\mathbf{x}_k - t\,\mathbf{g}_k), \tag{4.34}$$

where $\mathbf{g}_k$ is the objective gradient at the current iterate $\mathbf{x}_k$. The Cauchy point is obtained by successively examining the quadratic model (4.21) on each segment of (4.34). On a segment between two breakpoints, the model is a second-order polynomial function of the nonnegative parameter t. If the polynomial is nonincreasing on the segment, then the procedure moves to the next segment. Otherwise a minimizer is computed on the current segment and returned as the Cauchy point (Byrd *et al.*, 1995).

For clarity, we now drop the iteration index k. The Cauchy point $\mathbf{x}^{\mathrm{C}}$ yields the set of fixed indices

$$I_+^{\mathrm{C}} = \left\{ i \mid x_i^{\mathrm{C}} = 0 \text{ and } \partial_i f(\mathbf{x}) > 0 \right\}, \tag{4.35}$$

the associated subspace

$$F = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \forall\, i \in I_+^{\mathrm{C}} \quad x_i = x_i^{\mathrm{C}} \right\}, \tag{4.36}$$

and the active face $F_+ = F \cap \mathbb{R}^n_+$. In order to compute a minimizer of (4.21) over F and obtain a descent direction, we set $\mathbf{B} = \mathbf{B}_k$ and $\mathbf{g} = \mathbf{g}_k$ in (4.17) and solve by way of the Sherman-Morrison-Woodbury formula. If we partition

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_F \\ \mathbf{W}_G \end{pmatrix}, \tag{4.37}$$

where $\mathbf{W}$ is defined in (4.25), the inverse of a principal submatrix of $\mathbf{B}$ is

$$\mathbf{B}_{FF}^{-1} = \mathbf{B}_{0,FF}^{-1} - \mathbf{B}_{0,FF}^{-1}\mathbf{W}_F \left(\mathbf{M}^{-1} - \mathbf{W}_F^{\mathrm{T}}\mathbf{B}_{0,FF}^{-1}\mathbf{W}_F \right)^{-1} \mathbf{W}_F^{\mathrm{T}}\mathbf{B}_{0,FF}^{-1}. \tag{4.38}$$

With $\mathbf{B}_0 = \theta\mathbf{I}$, we have

$$\mathbf{B}_{FF}^{-1} = \theta^{-1}I - \theta^{-2}\mathbf{W}_F\left(\mathbf{M}^{-1} - \theta^{-1}\mathbf{W}_F^{\mathrm{T}}\mathbf{W}_F\right)^{-1}\mathbf{W}_F^{\mathrm{T}}. \tag{4.39}$$

Finally, we determine the next iterate by strong Wolfe linesearch between the current iterate and the solution of (4.17), subject to the bound constraints.

Algorithm 4.1 shows an overview of the L-BFGS-B method.

Algorithm 4.1: Overview of the standard L-BFGS-B algorithm.

Data: $\mathbf{x}_0$, parameters
for $k = 0, 1, 2, \ldots$ **do**
  Compute an exact Cauchy Point $\mathbf{x}_k^{\mathrm{C}}$ along the projected path
   $t \mapsto \mathrm{Proj}\left(\mathbf{x}_k - t\,\nabla f(\mathbf{x}_k)\right)$
  Identify the active face F_+
  Find a minimizer of the model (4.21) over the affine subspace (4.36) by the
   Sherman-Morrison-Woodbury formula
  Perform a strong Wolfe linesearch to find the next iterate $\mathbf{x}_{k+1}$
  Update the L-BFGS operator.
end

Implementation in Matlab

Our MATLAB implementation of L-BFGS-B [1], `lbfgsb.m`, solves (4.17) with CG instead of the Sherman-Morrison-Woodbury formula. Indeed, though `lbfgsb.m` uses (4.27) as an initial matrix, our implementation should work with (4.33) and (4.31). When $\mathbf{B}_0$ is nondiagonal, computing one of the products between $\mathbf{B}_{0,FF}^{-1}$ and a vector in (4.38) requires to solve a linear system and might be as expensive as computing a product with $\mathbf{B}_{FF}^{-1}$ by CG.

We validated our implementation against a C translation of the original Fortran implementation (Zhu *et al.*, 1997; Morales et Nocedal, 2011) provided by Stephen Becker [2] on a collection of standard problems. The C version uses the Sherman-Morrison-Woodbury formula to solve (4.17). Our benchmark comprises 128 bound-constrained problems from the CUTEst library (Gould, Orban, et Toint, 2015). The tests ran on a 3GHz Intel Core i7-5960X with 64GB of RAM. We report our results in the form of performance profiles in logarithmic scale (Dolan et Moré, 2002) in Figure 4.1.

Because we use of MATLAB instead of C, `lbfgsb.m` is slower than the original version. However, the results are similar in terms of number of objective evaluations and iterations,

1. Available online at `https://github.com/optimizers/NLPLab`
2. Available online at `https://github.com/stephenbeckr/L-BFGS-B-C`

even though a slight degradation in performance on standard problems is somewhat expected and matches the observations of Byrd *et al.* (1995).

Modifications related to scaling

In order to obtain a better Hessian approximation, we use (4.31) and (4.33) in the L-BFGS operator. A non-diagonal $\mathbf{B}_0$ requires several modifications in the algorithm, as the procedures presented by Byrd *et al.* (1995) owe their efficiency to the diagonal structure of $\mathbf{B}_0$.

Finding the Cauchy point requires examining up to n segments defined by the breakpoints along the projected gradient path. The feasibility of an exact search relies on the absence of any operation with complexity worse than $O(n)$ in the update of the quadratic model derivatives along each segment. In particular, the scalar product between a row of $\mathbf{B}_0$ and a vector is required for each segment visited. This operation is acceptable if applying $\mathbf{B}_0$ to a vector is cheap, and in particular for diagonal $\mathbf{B}_0$. In our case, applying $\mathbf{B}_0$ to a vector costs $O(n \log n)$ operations and quickly becomes time consuming. Instead, we use an Armijo-like backtracking search, similar to that implemented in TRON (Lin et Moré, 1999, Section 6). In the inexact procedure, the Cauchy step must only satisfy the sufficient decrease condition

$$m_k(\mathbf{x}^{\mathrm{C}}) \leqslant f(\mathbf{x}_k) + \mu_0\, \mathbf{g}^{\mathrm{T}}(\mathbf{x}^{\mathrm{C}} - \mathbf{x}_k), \tag{4.40}$$

where $0 < \mu_0 < \frac{1}{2}$.

Our implementation is summarized in Algorithm 4.2. In the next sections refer to it as `scaled-lbfgsb.m`.

Algorithm 4.2: Overview of the modified L-BFGS-B algorithm.

Data: $\mathbf{x}_0$, parameters
for $k = 0, 1, 2, \ldots$ **do**
  Identify the binding set $I_+(\mathbf{x}_k)$
  Compute an inexact Cauchy point $\mathbf{x}^{\mathrm{C}}$ along the projected path $t \mapsto \mathrm{Proj}\left(\mathbf{x}_k - t\,\bar{\mathbf{P}}\nabla f(\mathbf{x}_k)\right)$
  Identify the active face F_+
  Find a minimizer of the model (4.21) over the affine subspace (4.36) using preconditioned CG
  Perform a strong Wolfe linesearch to find the next iterate $\mathbf{x}_{k+1}$
  Update the L-BFGS operator.
end

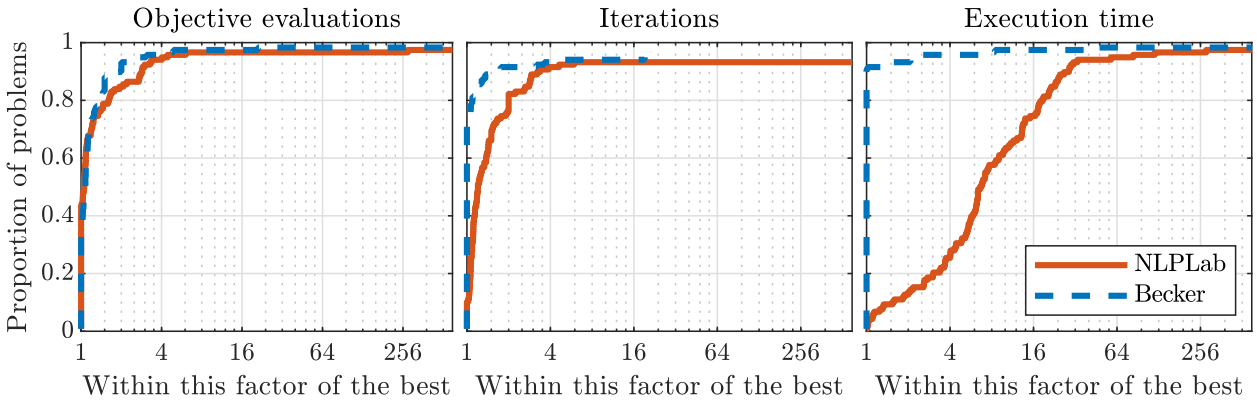

Figure 4.1 Log-scale performance profiles of our MATLAB implementation versus the C interface. The values compared are the number of objective evaluations (left), the number of iterations (middle) and the execution time (right).

4.3.2 A trust-region Newton method

Because second-order derivatives are available in our reconstruction problem, we also describe our scaling strategy in the context of a Newton method.

Presentation of the algorithm

TRON is a trust-region Newton method proposed by Lin et Moré (1999). As in L-BFGS-B, a general iteration includes the identification of an active face via a Cauchy point and the minimization of a quadratic model over an affine subspace corresponding to the free indices.

The quadratic model (4.21) now uses $\mathbf{B}_k = \nabla^2 f(\mathbf{x}_k)$. Due to the high cost of evaluating the quadratic model, we only compute an inexact Cauchy point satisfying (4.40).

Computating of a Newton direction requires finding an approximate solution of the trust-region subproblem

$$\begin{aligned} &\min \tfrac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{B}_k\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{g} \\ &\text{s.t. } \mathbf{x} \in F_+, \quad \|\mathbf{x} - \mathbf{x}_k\| \leqslant \Delta_k, \end{aligned} \tag{4.41}$$

where $\Delta_k > 0$ is the trust-region radius. We generate successive minor iterates $\mathbf{x}^j$ starting with $\mathbf{x}^0 = \mathbf{x}^{\mathrm{C}}$, that satisfy

$$m_k(\mathbf{x}^{j+1}) \leqslant m_k(\mathbf{x}^j) + \min(0,\, \mu_0\langle \mathbf{x}^{j+1} - \mathbf{x}^j, \nabla m_k(\mathbf{x}^j)\rangle). \tag{4.42}$$

To compute the first minor iterate $\mathbf{x}^1$, we solve (4.17) in the whole subspace $F^0 = F$ with CG. If $\hat{\mathbf{x}}$ is the solution and $\mathbf{z} = \hat{\mathbf{x}} - \mathbf{x}^{\mathrm{C}}$, we perform a projected search along the path

$t \mapsto \text{Proj}(\mathbf{x}^{\text{C}} + t\mathbf{z})$ to determine $\mathbf{x}^1$. Then, we set the new active face to

$$F^1 = \left\{ \mathbf{x} \in F^0 \mid x_i^1 = 0 \Rightarrow x_i = 0 \right\}, \tag{4.43}$$

and we launch CG again to minimize the quadratic model in F^1. The procedure terminates when a CG iterate falls outside the trust-region or when a minor iterate satisfies a sufficient decrease condition of type

$$\|\mathbf{x}_k - \text{Proj}(\mathbf{x}_k - \nabla f(\mathbf{x}_k))\| \leqslant \varepsilon \|\mathbf{x}^j - \text{Proj}(\mathbf{x}^j - \nabla m_k(\mathbf{x}^j))\|. \tag{4.44}$$

When one of those stopping conditions is met, the decrease of the quadratic model and the decrease of the objective function at the last minor iterate are compared. Depending on this information, the last minor iterate is accepted as $\mathbf{x}_{k+1}$ or rejected, and the trust-region radius is updated.

A summary of a TRON iteration is given in Algorithm 4.3.

Algorithm 4.3: Overview of the standard TRON algorithm.

Data: $\mathbf{x}_0$, parameters
for $k = 0, 1, 2, \ldots$ **do**
  Compute an inexact Cauchy Point $\mathbf{x}^{\text{C}}$ along the projected path
   $t \mapsto \text{Proj}\left(\mathbf{x}_k - t\,\nabla f(\mathbf{x}_k)\right)$
  Identify the active face F_+
  $j \leftarrow 0$, $F^0 \leftarrow F$, $\mathbf{x}^0 \leftarrow \mathbf{x}^{\text{C}}$
  while *(4.44) is not satified* **do**
    Compute a minimizer $\hat{\mathbf{x}}$ of the quadratic model over F^j subject to the
     trust-region constraint with CG
    Perform a projected search between $\mathbf{x}^j$ and $\hat{\mathbf{x}}$ to compute $\mathbf{x}^{j+1}$
    Update F^{j+1} with formula (4.43)
    $j \leftarrow j + 1$
  end
  Accept or reject $\mathbf{x}^j$ as the new iterate and update the trust-region radius
end

Implementation in Matlab

In the original TRON Fortran implementation, the conjugate gradient is preconditioned using an incomplete Cholesky factorization of $\mathbf{B}_k$. Such a factorization is not appropriate for large problems because the matrix coefficients are not explicitly available.

We use `Bcflash`, a Matlab implementation of TRON provided by Friedlander and Orban [3], without preconditioning and where $\mathbf{B}_k$ is only used as an operator.

For validation, we test `Bcflash` against the Fortran TRON implementation from which the incomplete Cholesky factorization was removed. The profiles in Figure 4.2 show the performance results on 127 problems from the CUTEst library (Gould *et al.*, 2015). The profiles show that `Bcflash` is more efficient in terms of function evaluations and Hessian products than the Fortran version. Moreover, the Matlab implementation is more robust. These results confirm the validity of Bcflash as an implementation of TRON.

`Bcflash` is competitive with the Fortran implementation in terms of execution time, whereas the difference is larger for L-BFGS-B. This can be partially explained as follows. In TRON, the bulk of the computation resides in Hessian-vector products. In both implementations the latter are computed by the CUTEst infrastructure, so this part of the computation is common between them. In L-BFGS-B, the objective function and gradient are only called at the beginning of the iteration and during the line search. Moreover, computations related to using and updating the limited-memory operator are difficult to vectorize efficiently, as they include operating on triangular matrices and reordering matrix columns. Thus, `lbfgsb.m` is at a disadvantage because those computations are implemented in Matlab.

A summary of the scaled variant of TRON appears in Algorithm 4.4. From here on, we refer to it as `scaled-Bcflash`.

Algorithm 4.4: Overview of the modified TRON algorithm.

Data: $\mathbf{x}_0$, parameters
for $k = 0, 1, 2, \ldots$ **do**
  Identify the binding set $I_+(\mathbf{x}_k)$
  Compute an inexact Cauchy Point $\mathbf{x}^{\mathrm{C}}$ along the projected path
  $t \mapsto \mathrm{Proj}\left(\mathbf{x}_k - t\,\bar{\mathbf{P}}\nabla f(\mathbf{x}_k)\right)$
  Identify the active face F_+
  $j \leftarrow 0$, $F^0 \leftarrow F$, $\mathbf{x}^0 \leftarrow \mathbf{x}^{\mathrm{C}}$
  while *(4.44) is not satified* **do**
    Compute a minimizer $\hat{\mathbf{x}}$ of the quadratic model over F^j subject to the trust-region constraint with CG preconditioned with $\mathbf{P}_{FF}$
    Perform a projected search between $\mathbf{x}^j$ and $\hat{\mathbf{x}}$ to compute $\mathbf{x}^{j+1}$
    Update F^{j+1} with formula (4.43)
    $j \leftarrow j + 1$
  end
  Accept or reject $\mathbf{x}^j$ as the new iterate and update the trust-region radius
end

3. Available online at `https://github.com/optimizers/NLPLab`

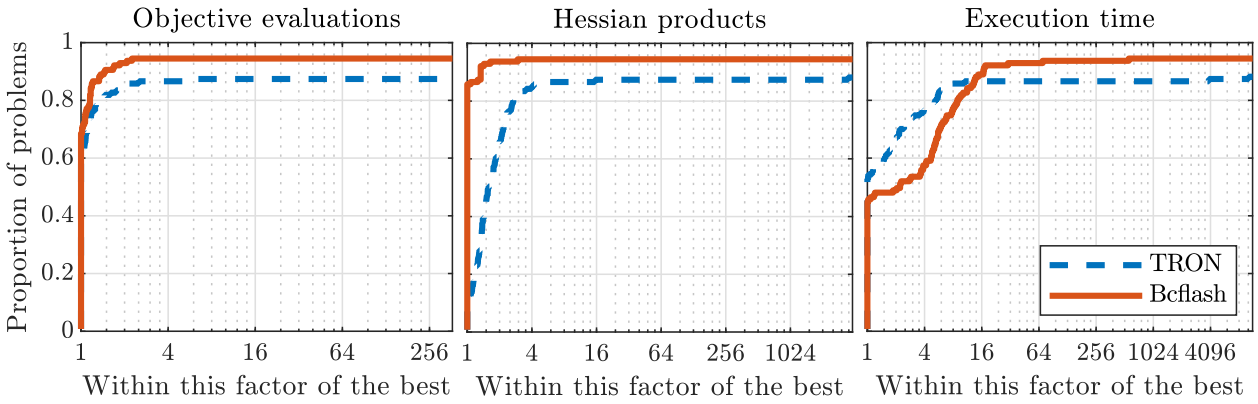

Figure 4.2 Performance profiles of Bcflash (Matlab) versus TRON without factorization (Fortran).

4.4 Numerical results

We now evaluate the performance of Algorithm 4.2 and Algorithm 4.4 on two image reconstruction problems. For both problems, we compare the performance of `scaled-lbfgsb.m` and `lbfgsb.m`, and that of `scaled-Bcflash` and `Bcflash`. We also compare `scaled-Bcflash` with a change-of-variable approach, by using `Cflash`, an implementation of TRON adapted to polyhedral constraints, to solve (4.4). In `Cflash`, projections are made onto the feasible set by solving a quadratic problem with Krylov methods (McLaughlin, 2017). Performances are compared in terms of projected gradient norm decrease and cumulated number of CG iterations along the reconstruction procedure.

First, to compare the convergence properties of the algorithms, we solve a simplified reconstruction problem, which is quadratic and better scaled than (4.2). In this first test, we also measure the fraction of execution time spent computing products with $\mathbf{A}$ and $\mathbf{C}$ for `Bcflash`, `scaled-Bcflash` and `Cflash`, in order to emphasize the high cost of constraints management in the third method compared to that in the two other methods.

In a second test, we use our methods on a real reconstruction problem. We evaluate the convergence acceleration caused by the use of scaled directions in this more complex case. To justify the choice of higher-order methods in image reconstruction, we also compare the performance of `scaled-Bcflash` with that of a first-order method, the spectral projected gradient (SPG) of Birgin et Martínez (2002).

In the following tests, we reconstruct images from a $672 \times 1 \times 1160$ synthetic sinogram with Poisson noise. In order to keep reasonable reconstruction times, we only consider 2D images. The data were created from a XCAT phantom (Segars, Mahesh, Beck, Frey, et Tsui, 2008)

of size 512×512 in cartesian coordinates. We reconstruct a discretized image using polar coordinates, with 226 radial subdivisions and 1160 angular subdivisions. This discretization provides a sufficient resolution to obtain, after conversion, a 512×512 cartesian image. Thus, the data creation and the image reconstruction are made using different procedures. In this problem, $\mathbf{A}$ has 779,520 rows and 262,160 columns, and the initial guess is $\mathbf{x}_0 = 0$.

All results below are produced on an Intel® Xeon® E5-2637 v4 processor at 3.50 GHz and 32 GB of RAM.

4.4.1 Simplified problem

We first consider the regularized linear least-squares problem

$$\min_{\mathbf{x} \geqslant 0} \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \tfrac{1}{2}\lambda\|\mathbf{K}\mathbf{x}\|^2, \tag{4.45}$$

where $\mathbf{K}\mathbf{x}$ models the differences between adjacent pixels of $\mathbf{x}$. In this simplified reconstruction problem, we drop the weight matrix $\mathbf{V}$, which is equivalent to assuming that all attenuation measures have the same variance, and we choose a L_2 regularization to keep the problem quadratic. We set $\lambda = 10^{-2}$ because it provides reasonable image quality and convergence speed.

Figure 4.3 shows the comparison between `lbfgsb.m` and `scaled-lbfgsb.m`. The left and right plots compare the decrease of the optimality residual and the cumulated number of CG iterations, respectively. We observe that the projected gradient norm decreases faster in the scaled case, especially in the first iterations, and that `scaled-lbfgsb.m` requires about half as many CG iterations per outer iteration as `lbfgsb.m`. The use of a nondiagonal $\mathbf{B}_0$ yields L-BFGS approximations that are closer to the problem Hessian and lead to better progress at each step. For this reason, we see on the left plot that `scaled-lbfgsb.m` decreases the projected gradient norm more than `lbfgsb.m` while doing less outer iterations. Moreover, each outer iteration has a lower cost in terms of CG iterations due to the use of a preconditioner when solving (4.17).

However, the performance of `scaled-lbfgsb.m` is not sufficient. Though much progress is achieved during the 30 first seconds, the convergence seems to switch to a linear behavior at some point, and it takes more than two minutes to decrease the projected gradient norm by a factor of 10^5.

Figure 4.4 reports corresponding results for TRON, where we compare `Bcflash`, `Cflash` and `scaled-Bcflash`. Both `Cflash` and `scaled-Bcflash` decrease the projected gradient much faster than `Bcflash`. Even though `scaled-Bcflash` requires more iterations than `Cflash`

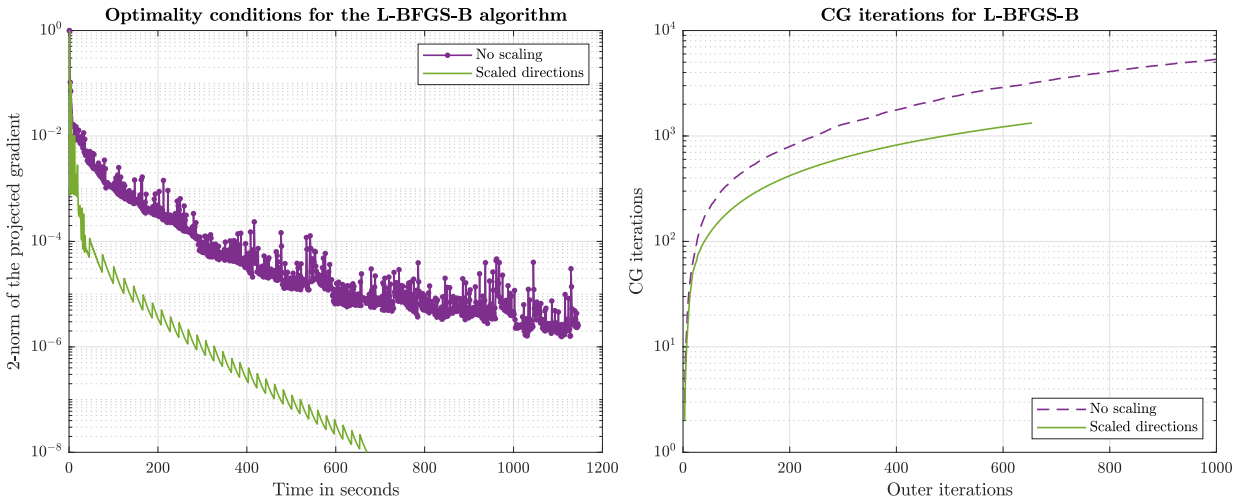

Figure 4.3 Convergence results for L-BFGS-B on (4.45).

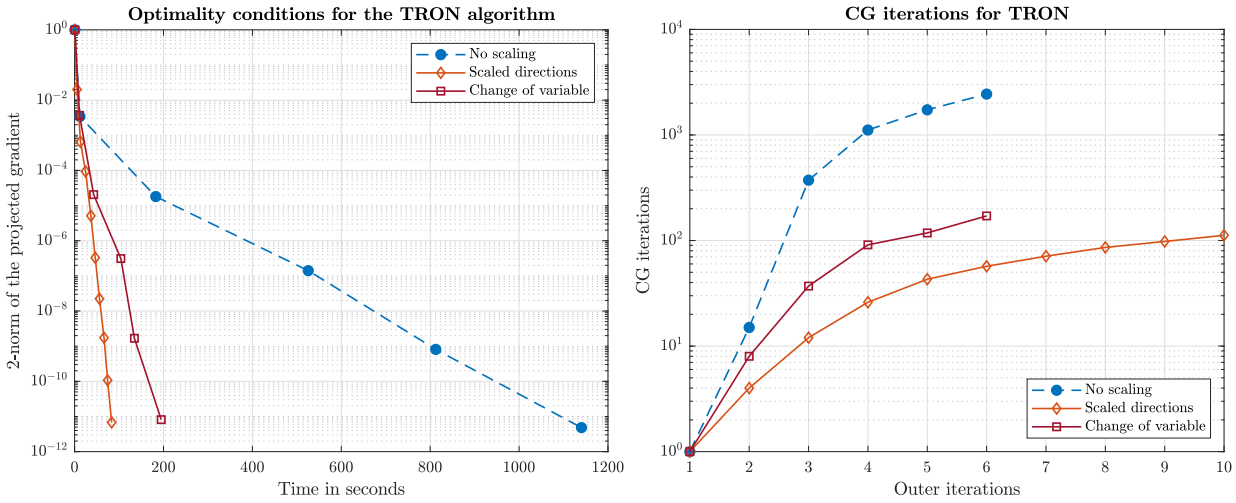

Figure 4.4 Convergence results for TRON on (4.45).

for the same gradient decrease, it converges faster.

Table 4.1 gathers some statistics about the execution of TRON. Due to the problem size, a significant part of the execution time is associated with products with $\mathbf{A}$ or its transpose. We see in the table that this time fraction is similar for `Bcflash` and `scaled-Bcflash`, both of which work in the original space, while it is much smaller for `Cflash`, which works in the scaled space. This difference is associated with the time spent computing orthogonal projections onto affine constraints in `Cflash`. Indeed, 23% of the solve time is spent computing products with $\mathbf{C}$ or its transpose, most of which are computed during orthogonal projections. In `scaled-Bcflash`, time is saved on projections while the cost of conjugate gradient iterations remains similar.

4.4.2 Reconstruction problem

In the second test, we compare Algorithm 4.2 and Algorithm 4.4 on the reconstruction problem

$$\min_{\mathbf{x} \geqslant 0} \tfrac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathbf{V}}^2 + \lambda \phi(\mathbf{K}\mathbf{x}), \tag{4.46}$$

where $\phi : \mathbf{q} \mapsto \sum_i \sqrt{\delta^2 + q_i^2}$ is an edge-preserving L_2/L_1 penalty with $\delta > 0$, and $\mathbf{V}$ is the statistical weight matrix defined in (4.2).

Problem (4.46) should be more difficult to solve than (4.45), even for scaled methods. Indeed, the addition of weights deteriorates the Hessian conditioning, and the penalty is not quadratic. The objective Hessian in (4.46) has the form

$$\nabla^2 f(\mathbf{x}) = \mathbf{A}^{\mathrm{T}} \mathbf{V} \mathbf{A} + \mathbf{K}^{\mathrm{T}} \mathbf{N}(\mathbf{K}\mathbf{x}) \mathbf{K}, \tag{4.47}$$

where $\mathbf{N}(\mathbf{K}\mathbf{x})$ is a diagonal matrix that depends on $\mathbf{K}\mathbf{x}$. This Hessian is not block-circulant and is less likely to be well approximated by a block-circulant $\mathbf{P}$ than that of (4.45).

We choose $\lambda = 10^{-4}$ and $\delta = 10^{-1}$. We determine these parameters by comparing the quality of reconstructed images for several values of λ and δ in terms of noise and blurring (Hamelin, 2009, section 4.5). To measure the noise level, we choose a zone that is uniform in the original

TRON variant	`Bcflash`	`scaled-Bcflash`	`Cflash`
Time fraction for $\mathbf{A}$-products	97 %	95 %	51 %
Time fraction for $\mathbf{C}$-products	0 %	1 %	23 %

Table 4.1 Execution statistics for the three versions of the TRON algorithm: fraction of time spent doing products with $\mathbf{A}$ and $\mathbf{C}$

phantom, and compare the variance of the corresponding pixels in the reconstructed images. To evaluate blurring, we assume the reconstructed image is obtained from the true phantom by a linear space-invariant process (Gonzalez et Woods, 2008, section 5.6.1). According to this model, the true phantom $\mathbf{x}_{\text{true}}$ and the reconstructed image $\mathbf{x}^*$ satisfy $\mathbf{x}^* = \mathbf{x}_{\text{true}} \star \mathbf{h} + \mathbf{n}$ where $\mathbf{h}$ is the point-spread function, $\mathbf{n}$ is a noise term and $\star$ is the 2D convolution operator. For each image, we use the width of $\mathbf{h}$ as a measure of blurring.

Figure 4.5 and Figure 4.6 show convergence results for L-BFGS-B and TRON, respectively. The solve time is longer than on (4.45) for all solvers, and the impact of the scaled solver is not as pronounced as in (4.45). The `scaled-lbfgsb.m` decreases the projected gradient by a factor of 10^4 about 9 times faster than `lbfgsb.m` on (4.45), but only 4 times faster on (4.46).

In the case of TRON, the advantage of `scaled-Bcflash` over `Cflash` is larger on (4.46) than on (4.45). The `scaled-Bcflash` decrease the projected gradient by a factor of 10^7 about 2.4 times faster than `Cflash` on (4.45), and 4.2 times faster on (4.46). So `scaled-Bcflash` is less affected by the scaling deterioration than `Cflash`.

Figure 4.7 shows a comparison between the convergence of `scaled-Bcflash` and the spectral projected gradient (SPG) of Birgin et Martínez (2002). SPG is sometimes used in imaging applications (Bonettini *et al.*, 2008) and is appealing because the cost of each iteration is low. We modify SPG to employ the same scaling strategy as Algorithm 4.2 and Algorithm 4.4. SPG decreases the objective function decrease faster than `scaled-Bcflash` in the first iterations, which is to be expected from a first-order method. However, after a few iterations, the SPG projected gradient norm decrease slowly, whereas the decrease rate is superlinear for TRON. With a 3 minute time limit, `scaled-Bcflash` decreases the projected gradient by a factor of 10^{10} while SPG only achieves a reduction of about 10^5. We conclude that `scaled-Bcflash` is a more appropriate for solving the reconstruction problem at tolerances stricter than 10^{-5}.

Figure 4.8 shows images obtained by solving (4.46) with SPG with tolerance ranging from 10^{-1} to 10^{-5} on the projected gradient norm, and the `scaled-Bcflash` reconstruction with tolerance 10^{-10}, which serves as reference. The picture obtained by SPG with tolerance 10^{-5} is the only one that does not present artifacts compared to the reference. This shows that in order to obtain good image quality, we should solve (4.46) with a tolerance stricter than or equal to 10^{-5}. We see in Figure 4.7 that `scaled-Bcflash` can achieve such accuracy faster than SPG.

These numerical results show that the scaling strategy brings the expected performance improvements for the problems we are interested in. In the case of TRON, this improvement is better than that we obtain with a change of variable, due to the high cost of orthogonal

projections in the second approach.

4.5 Conclusion

We presented a scaling strategy for bound-constrained problems inspired from a change of variable, and integrated it into two projected-directions algorithms, L-BFGS-B and TRON. Though, our strategy can be implemented into most projected-directions algorithms for large bound-constrained problems with little code modifications. In this paper, we adopted a practical point of view, as we gave details about the implementation of scaling for each subroutine of the algorithms, including the preconditioning of CG to solve quadratic subproblems that appear in higher-order methods. The numerical tests on badly scaled image reconstruction problems show that this approach gives better results than a change of variable, especially because the management of contraints is cheaper.

These results are promising for applications in X-Ray CT reconstruction, as they show the feasibility of reconstructing images in cylindrical coordinates. The partially diagonal scaling ensures the efficiency of the procedure, making fast and memory-saving reconstructions possible. In particular, TRON is a good candidate for applications which require to solve the reconstruction problem with a tight tolerance. As TRON can solve (4.46) with tolerance 10^{-10}, the reconstructed image is very close to the problem solution, and can be used as a reference to evaluate the convergence speed of other algorithms.

Here we have implemented the scaling into generic methods for large bound-constrained problems and solved generic reconstruction problems. In a future work, we will produce scaled methods for specific applications in medical image reconstruction, in order to combine the memory savings provided by the cylindrical coordinates with performance compatible with clinical applications. In particular, cylindrical coordinates are appropriate when the source and the detector follow a circular trajectory around the object to investigate, like in cone-beam computed tomography or in nondestructive testing. Structured system matrices can appear for other acquisition protocols, as long as the image discretisation yields geometrical invariances. Thus, block-circulant system matrices may also appear in helical computed tomography by using an helical discretization.

Circulant structures also appear in other imaging problems, for which our strategy can be applied. In general, our methods can prove to be useful in applications which lead naturally to non-diagonal scaling operators, like partial differential equations and optimal control, or when the optimization problem is too badly scaled for a diagonal scaling to be efficient.

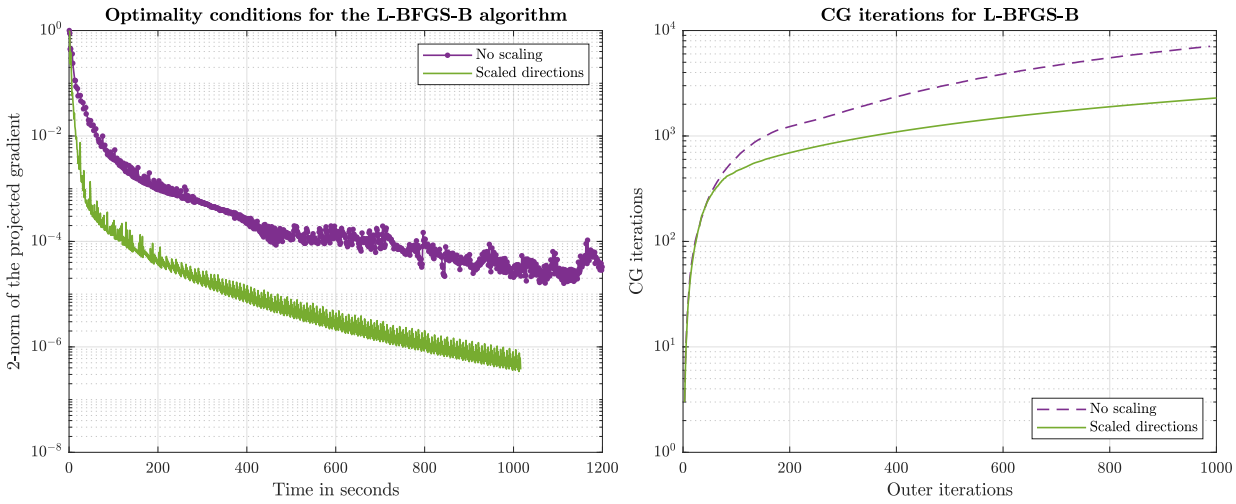

Figure 4.5 Convergence results for L-BFGS-B on (4.46)

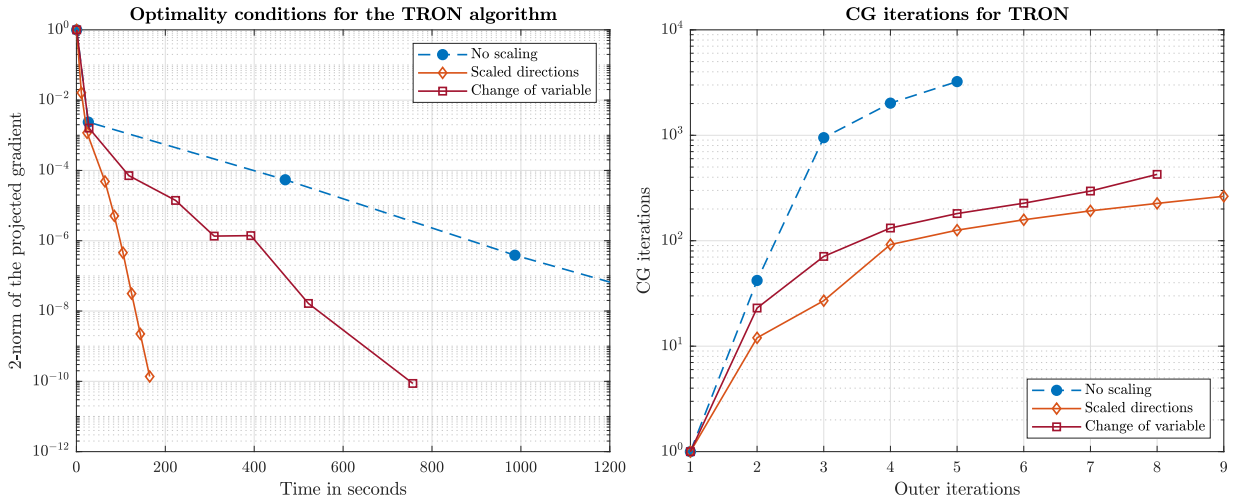

Figure 4.6 Convergence results for TRON on (4.46)

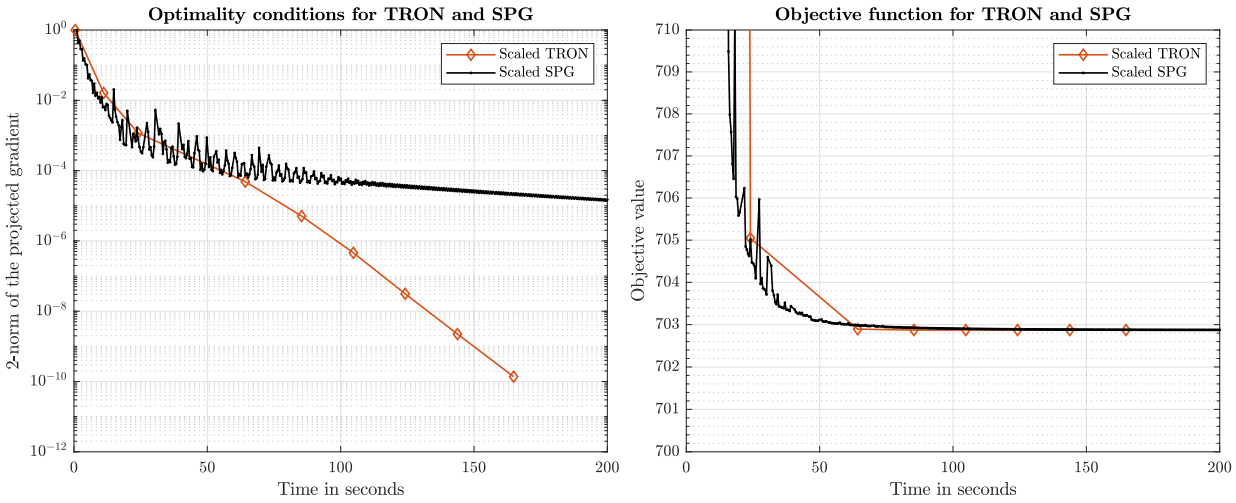

Figure 4.7 Comparison of TRON and SPG on (4.46)

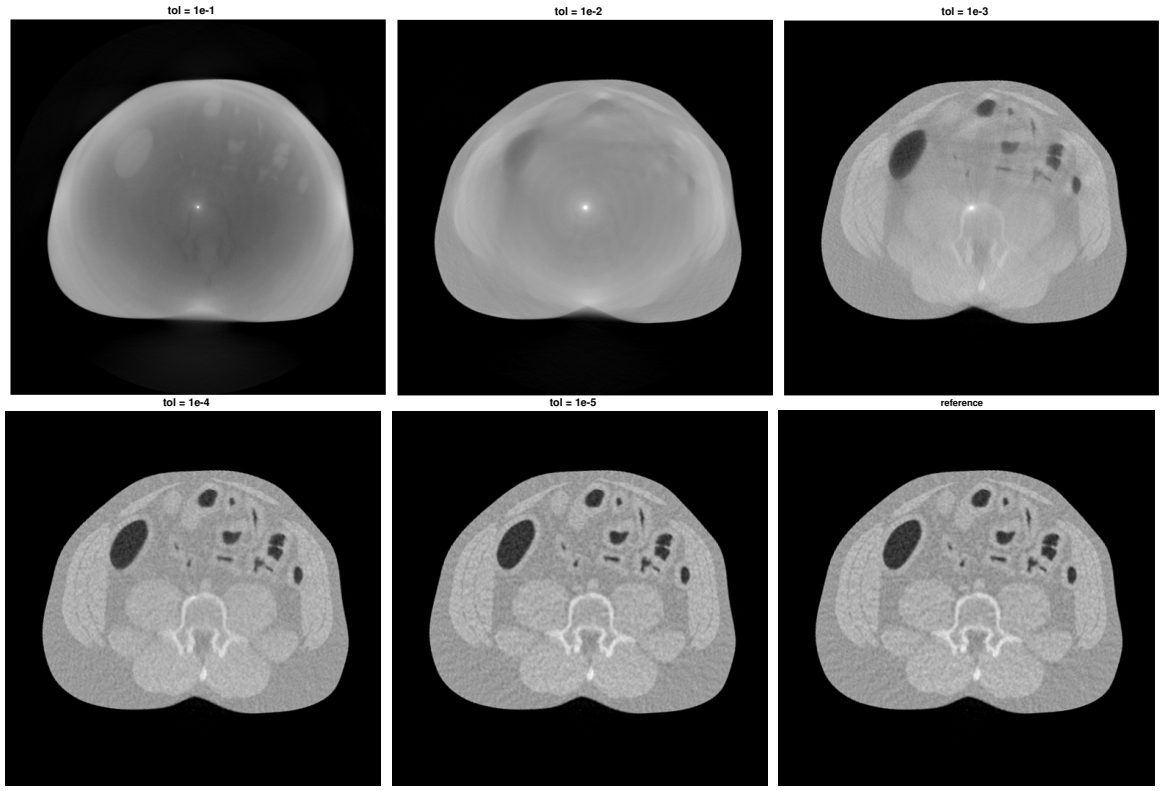

Figure 4.8 Images obtained with SPG for several tolerances. The reference image is obtained with TRON with tolerance 10^{-10}.

CHAPITRE 5 DISCUSSION GÉNÉRALE

Dans l'article présenté à l'annexe 4, nous résolvons efficacement le problème de reconstruction (1.2) en appliquant une mise à l'échelle aux méthodes de Newton et quasi-Newton que sont TRON et L-BFGS-B. Ce choix de directions mises à l'échelle semble naturel car il s'adapte bien aux contraintes de bornes, et par ailleurs il est beaucoup utilisé en imagerie, dans le cadre de méthodes de premier ordre.

Nous évoquons brièvement l'intérêt du travail effectué dans ce mémoire, après quoi nous discutons les choix réalisés pendant la maîtrise.

Portée et intérêt du travail Nous élargissons l'approche des directions mises à l'échelle à des méthodes plus complexes, une méthode de Newton et une méthode de quasi-Newton, et montrons que n'importe quelle méthode pour problème borné peut faire l'objet de la même accélération. La littérature en imagerie assimile souvent les méthodes d'ordre supérieur, comme la méthode de Newton projetée, à des méthodes de gradient auxquelles on applique une mise à l'échelle faisant intervenir la Hessienne ou une matrice de quasi-Newton. Notre étude est plus proche de la réalité du calcul, puisque nous partons d'une méthode d'ordre supérieur générique et nous détaillons l'application de la mise à l'échelle dans la génération de chaque direction. En particulier notre étude inclut le préconditionnement du gradient conjugué pouvant servir à inverser une matrice Hessienne.

La mise à l'échelle partiellement diagonale telle que nous la pratiquons s'applique bien en imagerie, car les structures circulantes y sont fréquentes et donnent lieu naturellement à un préconditionnement non-diagonal. Notre approche s'adresse aux applications pour lesquelles un préconditionneur non-diagonal apparaît naturellement, comme la résolution d'équations aux dérivées partielles. Par ailleurs, nos méthodes seront également utiles dès lors que le mauvais conditionnement d'un problème rend nécessaire l'utilisation d'un préconditionneur non-diagonal.

Enfin, l'article promeut l'utilisation en imagerie de méthodes d'optimisation génériques modernes. En particulier, la méthode TRON, grâce au calcul d'un point de Cauchy, est conçue spécialement pour traiter des grands problèmes, alors que la méthode de Newton projetée, préférée pour sa simplicité, peut s'avérer moins efficace. En cela, ce travail contribue au transfert des connaissances de la recherche en optimisation vers les applications.

Nous formulons maintenant quelques remarques sur certains détails de ce travail.

Choix des algorithmes Si la mise à l'échelle de TRON donne des résultats encourageants, on ne peut pas en dire autant de L-BFGS-B, dont le choix pour les essais de mise à l'échelle est discutable. D'une part, cette méthode se prête mal à la programmation en Matlab, car les opérations matricielles, surtout celles liées à la mise à jour L-BFGS, sont difficiles à vectoriser. D'autre part, la méthode doit son efficacité à de nombreuses astuces de programmation, et il est difficile de la modifier sans la dénaturer. Par exemple, on a dû modifier la procédure de recherche du point de Cauchy pour s'adapter à la mise à l'échelle. Les résultats d'essais montrent que notre mise en œuvre de L-BFGS-B est plus performante quand on diminue le nombre de paires. Pour cette raison, nous pensons qu'un méthode plus simple comme un gradient conjugué non-linéaire (Fletcher et Reeves, 1964; Polak et Ribière, 1969) pourrait mieux se comporter qu'une méthode de quasi-Newton, et serait plus simple à mettre en œuvre.

Influence de la taille du problème L'article montre la différence de vitesse de convergence entre l'approche par directions mises à l'échelle et celle utilisant le changement de variable, et il est établi qu'une partie conséquente du temps de reconstruction dans le cas avec changement de variable est liée au calcul des projections. Cependant, on ne sait pas comment évolue ce rapport quand on fait varier la taille du problème. La matrice de mise à l'échelle C s'applique en $\mathrm{O}(n \log n)$, tandis que la Hessienne du problème s'applique en $\mathrm{O}(n^2)$. Le coût d'un produit par C devient négligeable devant n^2 quand le problème grandit, mais on ne sait pas combien d'itérations sont nécessaires pour calculer une projection orthogonale. Il est donc difficile de dire comment évolue la part de temps de calcul dédié au calcul des projections, même si la simplicité de l'utilisation de directions mises à l'échelle reste un avantage dans le cas des problèmes de grande taille. Des travaux futurs devront éclaircir ce point.

Choix de la mise à l'échelle On utilise dans cette étude une matrice C conçue avec l'idée de faire un changement de variable. Dans le cadre des directions mises à l'échelle, nous ne sommes pourtant pas obligés d'utiliser une matrice C constante. Si le terme de moindres carrés possède une hessienne constante, le terme de pénalisation possède une hessienne variable et cela peut être pris en compte dans la mise à l'échelle. La mise à jour de C à chaque itération représente un coût supplémentaire, mais pourrait être bénéfique pour la convergence, par exemple dans le cas L_2L_1 où la hessienne de la pénalisation varie beaucoup. Une autre question à se poser concerne la validité du préconditionnement quand on extrait une sous-matrice principale de la matrice C pour effectuer la mise à l'échelle dans une face de l'ensemble réalisable. La matrice C dont nous disposons est basée sur la transformée de

Fourier discrète, qui, on le sait, est loin d'être une transformation locale. On peut donc se demander à quel point les vertus de la mise à l'échelle sont dégradées pour les sous-matrices de C. D'autres choix de transformations plus locales pourraient être envisagées pour avoir un meilleur comportement sur les sous-espaces. Par exemple, indépendemment des performances, une matrice tridiagonale serait sans doute plus en cohérence avec l'extraction de sous-matrices. De manière générale, le choix des opérateurs de mise à l'échelle est un sujet qu'il est souhaitable d'approfondir.

CHAPITRE 6 CONCLUSION

6.1 Synthèse des travaux

Aux questions soulevées dans les travaux précédents quant au traitement des contraintes affines dans le cadre du problème (1.16), nous proposons une réponse basée sur une mise à l'échelle du problème (1.2) au lieu d'un changement de variable. La mise à l'échelle, choisie à chaque itération, prend la forme d'un changement de métrique dans le calcul des directions de descente et des projections orthogonales. Grâce à un choix de métrique qui dépend des contraintes actives au point où l'on se trouve, on montre qu'il est possible de calculer les projections au moyen d'une formule directe. On élimine donc le besoin de méthodes itératives pour calculer les projections.

Au-delà de l'abandon du changement de variable, qui est juste une question de forme, le résultat à retenir de ce mémoire est l'intérêt d'utiliser une métrique adaptative, et plus généralement l'importance d'attacher de l'attention au choix de la métrique à chaque itération. Ainsi, le succès de la reconstruction passe par le choix d'une méthode qui correspond à l'application, par exemple la tolérance nécessaire, et, parallèlement, au choix d'une métrique dans laquelle le problème a des bonnes propriétés numériques.

Ces résultats constituent sans doute une contribution significative au projet de reconstruction tomographique en coordonnées cylindriques. En effet, on montre ici la faisabilité de la reconstruction d'image au moyen de méthodes projetées, et on donne de nouvelles perspectives pour accélérer la convergence.

6.2 Limitations de la solution proposée

En nous concentrant sur la mise à l'échelle du problème de reconstruction, nous nous sommes éloignés du projet dans lequel cette maîtrise s'insère. Rappelons que la principale application visée est de promouvoir l'usage des coordonnées cylindriques en tomographie médicale par rayons X. Pour cela, il est nécessaire de montrer à la communauté scientifique qu'on peut, en utilisant les coordonnées cylindriques, obtenir des images aussi rapidement et présentant une aussi bonne qualité qu'en coordonnées cartésiennes, et ce en réalisant d'importants gains en mémoire et en temps de calcul.

Notre étude constitue donc un pas vers cet objectif mais plusieurs détails restent à régler, car elle ne tient pas compte de la réalité des applications en tomographie médicale.

Par exemple, tous nos essais ont été faits en deux dimensions, c'est-à-dire qu'on ne reconstruit qu'une tranche du patient, et qu'on suppose que le faisceau de rayons X est dans le plan de l'image. Les reconstructions dans la pratique se font en trois dimensions. Pour certaines modalités d'imagerie, la géométrie en coordonnées cylindriques est adaptée, mais ce n'est pas toujours le cas. En particulier, l'adaptation aux acquisitions à balayage hélicoïdal n'a pas été considérée ici.

De, même, le choix de méthodes relativement complexes a permis d'illustrer l'intégration de la mise à l'échelle dans les différentes procédures ainsi que les compromis à réaliser. En revanche, dans le cadre de la reconstruction tomographique, il n'est pas toujours nécessaire de résoudre le problème à une tolérance 10^{-10}. Pour les applications génériques les plus courantes, les tolérances sont larges, et les méthodes de premier ordre sont de rigueur. Seules des applications spécifiques nécessitant une bonne exploitation des données disponibles nécessitent de résoudre le problème strictement et justifient l'utilisation de méthodes d'ordre supérieur.

Enfin, du fait de l'usage de solveurs génériques, la métrique utilisée pour évaluer la convergence des algorithmes est la norme du gradient projeté, une mesure générique en optimisation. Comme nous ne discutons pas de la qualité des images obtenues, ou de leurs caractéristiques par rapport à une appliation donnée, il est encore difficile de comparer notre approche à l'état de l'art en imagerie.

Nous avons progressé sur la résolution du problème d'optimisation. Reconstruire des images avec une tolérance stricte est utile pour étudier les caractéristiques du problème d'optimisation et de ses solutions. Néanmoins on est encore loin d'une application réelle en reconstruction d'image.

6.3 Améliorations futures

Comme on l'a dit en section 6.2, l'avancement du projet nécessite de garder en tête les objectifs à long terme. La suite du travail doit sans doute être orientée vers des applications médicales bien identifiées, afin de se placer par rapport à la littérature récente.

Le but, encore une fois, est de convaincre la communauté de l'intérêt des coordonnées cylindriques. Cela nécessite la réalisation d'essais numériques contre l'état de l'art, par exemple l'algorithme OS-SQS (Erdoğan et Fessler, 1999b) et ses variantes, sur des problèmes à données réelles. Dans ce cas, les coordonnées cylindriques associées à la mise à l'échelle non-diagonale seront à comparer à la mise à l'échelle diagonale en coordonnées cartésiennes. Des publications devront créer une base permettant de progresser dans le projet.

Une reconstruction performante passe par le développement d'algorithmes spécifiques, dans des langages bas niveaux, et faisant usage des mêmes accélérations que les algorithmes existants : accélération de Nesterov, sous-ensembles ordonnés, parallélisation, etc.

De plus, c'est à nos méthodes de s'adapter aux conditions cliniques et pas l'inverse. Ainsi, la plupart des travaux réalisés jusque maintenant font l'hypothèse $V = I$ dans le problème (1.2). En particulier, Golkar (2013) conçoit la matrice de mise à l'échelle en se basant sur le problème simplifié. À l'avenir il faudra se confronter au vrai problème de reconstruction même s'il est plus difficile à traiter, sans quoi il sera impossible d'être crédible dans la littérature.

Enfin, au-delà des coordonnées cylindriques, le projet consiste à utiliser une discrétisation adaptée à la géométrie de l'acquisition des données. Une telle idée peut probablement être appliquée à d'autres géométries, comme les pixels hélicoïdaux. Encore une fois, il est nécessaire de se rapprocher des applications cliniques pour trouver une discrétisation adaptée à chaque protocole d'acquisition.

En conclusion, la suite du projet devrait concrétiser les études amont réalisées jusque maintenant en s'attaquant à l'application dans des conditions réelles.

RÉFÉRENCES

S. Ahn, J. A. Fessler, D. Blatt et A. O. Hero, III : Convergent incremental optimization transfer algorithms : Application to tomography. *IEEE Trans. Medical Imaging*, 25(3):283–296, Mars 2006.

J. Barzilai et J. M. Borwein : Two-point step size gradient methods. *IMA J. Num. Anal.*, 8(1):141–148, 1988.

A. Beck : *First-Order Methods in Optimization*, vol. 25. SIAM, 2017.

A. Beck et M. Teboulle : A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J.Imaging Siences*, 2(1):183–202, 2009.

D. P. Bertsekas : On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automat. Contr.*, 21(2):174–184, avr. 1976.

D. P. Bertsekas : Projected Newton methods for optimization problems with simple constraints. *SIAM J. Control Optimization*, 20(2):221–246, 1982.

E. G. Birgin et J. M. Martínez : Large-scale active-set box-constrained optimization method with spectral projected gradients. *Comput. Optim. Appl.*, 23:101–125, 2002.

S. Bonettini, G. Landi, E. L. Piccolomini et L. Zanni : Scaling techniques for gradient projection-type methods in astronomical image deblurring. *International Journal of Computer Mathematics*, 90(1):9–29, 2013.

S. Bonettini, R. Zanella et L. Zanni : A scaled gradient projection method for constrained image deblurring. *Inverse Probl.*, 25(1):015002, 2008.

S. Boyd et L. Vandenberghe : *Convex Optimization*. Cambridge University Press, 2004.

J. V. Burke et J. J. Moré : Exposing constraints. *SIAM J. Optimization*, 4(3):573–595, 1994.

R. H. Byrd, P. Lu, J. Nocedal et C. Zhu : A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208, 1995.

R. H. Byrd, J. Nocedal et R. B. Schnabel : Representations of quasi-Newton matrices and their use in limited memory methods. *Math. Program.*, 63(1-3):129–156, 1994.

A. Chambolle et T. Pock : A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vision*, 40(1):120–145, 2011.

K. Chen : *Matrix preconditioning techniques and applications*, vol. 19 de *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, 2005.

K. Choi, J. Wang, L. Zhu, T. Suh, S. Boyd et L. Xing : Compressed sensing based cone-beam computed tomography reconstruction with a first-order method. *Math. Program.*, 37(9):5113–5125, 8 2010. ISSN 2473-4209.

L. Condat : A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2):460–479, Août 2013.

A. R. Conn, N. I. Gould et P. L. Toint : Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Num. Anal.*, 25(2):433–460, 1988.

A. R. Conn, N. Gould, A. Sartenaer et P. L. Toint : Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM J. Optimization*, 3(1):164–221, 1993.

J. E. Dennis, Jr et J. J. Moré : Quasi-Newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

E. D. Dolan et J. J. Moré : Benchmarking optimization software with performance profiles. *Math. Program. A*, 91:201–213, 2002.

H. Erdoğan et J. A. Fessler : Monotonic algorithms for transmission tomography. *IEEE Trans. Medical Imaging*, 18(9):801–814, Sep. 1999a.

H. Erdoğan et J. A. Fessler : Ordered subsets algorithms for transmission tomography. *Phys. Med. Biol.*, 44(11):2835–2851, Nov. 1999b.

L. A. Feldkamp, L. C. Davis et J. W. Kress : Practical cone-beam algorithm. *J. Opt. Soc. Am. (A)*, 1(6):612–619, Juin 1984.

J. A. Fessler : Statistical image reconstruction methods for transmission tomography. *In* J. M. Fritzpatrick et M. Sonka, éds : *Handbook of Medical Imaging*, vol. 2, chap. 1, p. 1–70. SPIE Press, Bellingham, WA, 2000.

R. Fletcher et C. M. Reeves : Function minimization by conjugate gradients. *Comput. J.*, 7(2):149–157, 1964.

E. M. Gafni et D. P. Bertsekas : Two-metric projection methods for constrained optimization. *SIAM J. Control Optimization*, 22(6):936–964, 1984.

L. L. Geyer, U. J. Schoepf, F. G. Meinel, J. W. Nance Jr, G. Bastarrika, J. A. Leipsic, N. S. Paul, M. Rengo, A. Laghi et C. N. De Cecco : State of the art : iterative CT reconstruction techniques. *Radiology*, 276(2):339–357, 2015.

L. W. Goldman : Principles of CT and CT technology. *J. Nucl. Med. Technol.*, 35(3):115–128, 2007.

M. Golkar : *Fast iterative reconstruction in X-ray tomography using polar coordinates*. Thèse de doctorat, École Polytechnique de Montréal, 2013.

R. C. Gonzalez et R. E. Woods : *Digital Image Processing*. Perarson Education, Inc., Upper Saddle River, NJ, 3rd édn, 2008.

N. I. Gould, D. Orban et P. L. Toint : CUTEst : A constrained and unconstrained testing environment with safe threads for mathematical optimization. *Comput. Optim. Appl.*, 60 (3):545–557, 2015.

Y. Goussard, M. Golkar, A. Wagner et M. Voorons : Cylindrical coordinate representation for statistical 3D CT reconstruction. *In Proc. Int. Meeting on Fully 3D Image Reconstr. in Rad. and Nucl. Med.*, p. 138–141, Lake Tahoe, CA, Juin 2013.

B. Hamelin : *Accélération d'une Approche Régularisée de Reconstruction en Tomographie à Rayons X avec Réduction des Artéfacts Métalliques*. Thèse de doctorat, École Polytechnique de Montréal, 2009.

B. Hamelin, Y. Goussard et J.-P. Dussault : Comparison of optimization techniques for regularized statistical reconstruction in X-ray tomography. *In Proc. Int. Conf. Image Proc. Theory, Tools and Appl.*, Paris, France, Juil. 2010a. 5 pages.

B. Hamelin, Y. Goussard, J.-P. Dussault, G. Cloutier, G. Beaudoin et G. Soulez : Design of iterative ROI transmission tomography reconstruction procedures and image quality analysis. *Med. Phys.*, 37(9):4577–4589, Sep. 2010b.

G. T. Herman : *Fundamentals of Computerized Tomography : Image Reconstruction From Projections*. Springer Science & Business Media, 2009.

M. R. Hestenes : Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4(5):303–332, 1969.

M. R. Hestenes et E. Stiefel : *Methods of conjugate gradients for solving linear systems*, vol. 49. NBS Washington, DC, 1952.

H. M. Hudson et R. S. Larkin : Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans. Medical Imaging*, 13(4):601–609, Déc. 1994.

T. L. Jensen, J. H. Jørgensen, P. C. Hansen et S. H. Jensen : Implementation of an optimal first-order method for strongly convex total variation regularization. *BIT Num. Math.*, 52(2):329–356, Juin 2012. ISSN 1572-9125.

M. Kaplan, D. Haynor et H. Vija : A differential attenuation method for simultaneous estimation of spect activity and attenuation distributions. *IEEE Transactions on Nuclear Science*, 46(3):535–541, 1999.

D. Kim, S. Ramani et J. A. Fessler : Combining ordered subsets and momentum for accelerated x-ray ct image reconstruction. *IEEE transactions on medical imaging*, 34(1): 167–178, 2014.

D. Kim, S. Sra et I. S. Dhillon : Tackling box-constrained optimization via a new projected quasi-Newton approach. *SIAM J. Sci. Comput.*, 32(6):3548–3563, 2010.

S. Labouesse, M. Allain, J. Idier, S. Bourguignon, A. Negash, P. Liu et A. Sentenac : Joint reconstruction strategy for structured illumination microscopy with unknown illuminations. *IEEE Trans. Signal Process.*, SP-26(5):2480–2493, Mai 2017.

G. Landi et E. Loli Piccolomini : A projected Newton-CG method for nonnegative astronomical image deblurring. *Numerical Algorithms*, 48(4):279–300, août 2008. ISSN 1572-9265.

K. Lange et J. A. Fessler : Globally convergent algorithms for maximum a posteriori transmission tomography. *IEEE Trans. Image Process.*, 4(10):1430–1438, Oct. 1995.

C.-J. Lin et J. J. Moré : Newton's method for large bound-constrained optimization problems. *SIAM J. Optimization*, 9(4):1100–1127, 1999.

M. McLaughlin : Méthodes sans factorisation pour la tomographie à rayons X en coordonnées cylindriques. Mémoire de D.E.A., École Polytechnique de Montréal, 2017. URL `https://publications.polymtl.ca/2742`.

J. L. Morales et J. Nocedal : Remark on "Algorithm 778 : L-BFGS-B : Fortran subroutines for large-scale bound constrained optimization". *ACM Transactions on Mathematical Software (TOMS)*, 38(1):7, 2011.

J. J. Moré : Recent developments in algorithms and software for trust region methods. *In Mathematical programming The state of the art*, p. 258–287. Springer, 1983.

J. J. Moré et G. Toraldo : On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optimization*, 1(1):93–113, 1991.

Y. E. Nesterov : A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *In Dokl. Akad. Nauk SSSR*, vol. 269, p. 543–547, 1983.

H. Nien et J. A. Fessler : Fast X-ray CT image reconstruction using a linearized augmented Lagrangian method with ordered subsets. *IEEE Transactions on Medical Imaging*, 34 (2):388–399, Fév. 2015. ISSN 0278-0062.

J. Nocedal : Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35 (151):773–782, 1980.

J. Nocedal et S. J. Wright : *Numerical Optimization*. Operations Research. Springer, New York, NY, 2006.

F. Noo, K. Hahn, H. Schöndube et K. Stierstorfer : Iterative ct reconstruction using coordinate descent with ordered subsets of data. *In Medical Imaging 2016 : Physics of Medical Imaging*, vol. 9783, p. 97834A. International Society for Optics and Photonics, 2016.

N. Parikh et S. Boyd : Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):123–231, 2013.

T. Pock et A. Chambolle : Diagonal preconditioning for first order primal-dual algorithms in convex optimization. *In Int. Conf. on Comp. Vision*, p. 1762–1769. IEEE, 2011.

E. Polak et G. Ribière : Note sur la convergence de méthodes de directions conjuguées. 3 (1):35–43, 1969.

S. Ravishankar, J. C. Ye et J. A. Fessler : Image reconstruction : From sparsity to data-adaptive methods and machine learning. *arXiv preprint arXiv :1904.02816*, 2019.

K. D. Sauer et C. A. Bouman : A local update strategy for iterative reconstruction from projections. *IEEE Trans. Signal Process.*, SP-41(2):534–548, Fév. 1993.

W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey et B. M. W. Tsui : Realistic CT simulation using the 4D XCAT phantom. *Med. Phys.*, 35(8):3800–8, Sep. 2008. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2809711/pdf/MPHYA6-000035-003800_1.pdf`.

L. A. Shepp et Y. Vardi : Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Medical Imaging*, MI-1(2):113–122, Oct. 1982.

E. Y. Sidky, J. S. Jørgensen et X. Pan : First-order convex feasibility algorithms for X-ray CT. *Med. Phys.*, 40(3), 2013.

W. Su, S. Boyd et E. Candes : A differential equation for modeling Nesterov's accelerated gradient method : Theory and insights. *In Advances in Neural Information Processing Systems*, p. 2510–2518, 2014.

C. Thibaudeau, J.-D. Leroux, R. Fontaine et R. Lecomte : Fully 3D iterative CT reconstruction using polar coordinates. *Med. Phys.*, 40(11):111904, 2013.

Q. Xu, D. Yang, J. Tan, A. Sawatzky et M. A. Anastasio : Accelerated fast iterative shrinkage thresholding algorithms for sparsity-regularized cone-beam ct image reconstruction. *Medical physics*, 43(4):1849–1872, 2016.

Y. Xu : Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM J. Optimization*, 27(3):1459–1484, 2017.

H. Zhang, J. Wang, D. Zeng, X. Tao et J. Ma : Regularization strategies in statistical image reconstruction of low-dose X-ray CT : A review. *Med. Phys.*, 45(10):e886–e907, 2018.

C. Zhu, R. H. Byrd, P. Lu et J. Nocédal : Algorithm 778. L-BFGS-B : Fortran subroutines for large-scale bound constrained optimization. *ACM Trans. Math. Soft.*, 23(4):550–560, 1997.