Proximal and Interior Point Optimization Strategies in Image Recovery

Marie-Caroline Corbineau

université PARIS-SACLAY

CentraleSupélec

NNT : 20XXSACLXXXX

Thèse de doctorat

Proximal and Interior Point Optimization Strategies in Image Recovery

Thèse de doctorat de l'Université Paris-Saclay
préparée à **CentraleSupélec**

École doctorale n° 580 Sciences et Technologies de l'Information et de la Communication (STIC)

Spécialité de doctorat : **Mathématiques & Informatique**

Thèse présentée et soutenue à Gif-sur-Yvette, le 3 décembre 2019, par

Marie-Caroline Corbineau

Composition du Jury :

François Malgouyres — Rapporteur
Professeur, Université Paul Sabatier, Institut de Mathématiques de Toulouse (Toulouse, France)

Silvia Villa — Rapporteur
Associate Professor, Università degli Studi di Genova, Dipartimento di Matematica (Gênes, Italie)

Denis Kouamé — Examinateur
Professeur, Université Paul Sabatier, IRIT (UMR CNRS 5505) (Toulouse, France).

Saïd Moussaoui — Examinateur
Professeur, Ecole Centrale de Nantes, LS2N (UMR CNRS 6004) (Nantes, France).

Lenka Zdeborová — Examinatrice
Chargée de recherche CNRS, Universitée Paris-Saclay, CEA, IPhT (UMR CNRS 3681) (Gif-sur-Yvette, France)

Émilie Chouzenoux — Co-Directrice de Thèse
Chargée de recherche Inria, Universitée Paris-Saclay, CentraleSupélec, Inria, CVN (Gif-sur-Yvette, France)

Jean-Christophe Pesquet — Directeur de Thèse
Professeur, Universitée Paris-Saclay, CentraleSupélec, Inria, CVN (Gif-sur-Yvette, France)

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Stratégies d'Optimisation Proximales et de Points Intérieurs en Reconstruction d'Images

Mots clés : algorithmes proximaux, points intérieurs, algorithme déroulé, imagerie hyperspectrale, imagerie ultrasonore, algorithme de Langevin

Résumé : Les problèmes inverses en traitement d'images peuvent être résolus en utilisant des méthodes variationnelles classiques, des approches basées sur l'apprentissage profond, ou encore des stratégies bayésiennes. Bien que différentes, ces approches nécessitent toutes des algorithmes d'optimisation efficaces. L'opérateur proximal est un outil important pour la minimisation de fonctions non lisses. Dans cette thèse, nous illustrons la polyvalence des algorithmes proximaux en les introduisant dans chacune des trois méthodes de résolution susmentionnées.
Tout d'abord, nous considérons une formulation variationnelle sous contraintes dont la fonction objectif est composite. Nous développons PIPA, un nouvel algorithme proximal de points intérieurs permettant de résoudre ce problème. Dans le but d'accélérer PIPA, nous y incluons une métrique variable. La convergence de PIPA est prouvée sous certaines conditions et nous montrons que cette méthode est plus rapide que des algorithmes de l'état de l'art au travers de deux exemples numériques en traitement d'images.
Dans une deuxième partie, nous étudions iRestNet, une architecture neuronale obtenue en déroulant un algorithme proximal de points intérieurs. iRestNet nécessite l'expression de l'opérateur proximal de la barrière logarithmique et des dérivées premières de cet opérateur. Nous fournissons ces expressions pour trois types de contraintes. Nous montrons ensuite que sous certaines conditions, cette architecture est robuste à une perturbation sur son entrée. Enfin, iRestNet démontre de bonnes performances pratiques en restauration d'images par rapport à une approche variationnelle et à d'autres méthodes d'apprentissage profond.
La dernière partie de cette thèse est consacrée à l'étude d'une méthode d'échantillonnage stochastique pour résoudre des problèmes inverses dans un cadre bayésien. Nous proposons une version accélérée de l'algorithme proximal de Langevin non ajusté, baptisée PP-ULA. Cet algorithme est incorporé à un échantillonneur de Gibbs hybride utilisé pour réaliser la déconvolution et la segmentation d'images ultrasonores. PP-ULA utilise le principe de majoration-minimisation afin de gérer les distributions non log-concaves. Comme le montrent nos expériences réalisées sur des données ultrasonores simulées et réelles, PP-ULA permet une importante réduction du temps d'exécution tout en produisant des résultats de déconvolution et de segmentation très satisfaisants.

Title : Proximal and Interior Point Optimization Strategies in Image Recovery

Keywords : proximal algorithms, interior points, unfolded algorithm, hyperspectral imaging, ultrasound imaging, Langevin-based schemes

Abstract : Inverse problems in image processing can be solved by diverse techniques, such as classical variational methods, recent deep learning approaches, or Bayesian strategies. Although relying on different principles, these methods all require efficient optimization algorithms. The proximity operator appears as a crucial tool in many iterative solvers for nonsmooth optimization problems. In this thesis, we illustrate the versatility of proximal algorithms by incorporating them within each one of the aforementioned resolution methods.
First, we consider a variational formulation including a set of constraints and a composite objective function. We present PIPA, a novel proximal interior point algorithm for solving the considered optimization problem. This algorithm includes variable metrics for acceleration purposes. We derive convergence guarantees for PIPA and show in numerical experiments that it compares favorably with state-of-the-art algorithms in two challenging image processing applications.
In a second part, we investigate a neural network architecture called iRestNet, obtained by unfolding a proximal interior point algorithm over a fixed number of iterations. iRestNet requires the expression of the logarithmic barrier proximity operator and of its first derivatives, which we provide for three useful types of constraints. Then, we derive conditions under which this optimization-inspired architecture is robust to an input perturbation. We conduct several image deblurring experiments, in which iRestNet performs well with respect to a variational approach and to state-of-the-art deep learning methods.
The last part of this thesis focuses on a stochastic sampling method for solving inverse problems in a Bayesian setting. We present an accelerated proximal unadjusted Langevin algorithm called PP-ULA. This scheme is incorporated into a hybrid Gibbs sampler used to perform joint deconvolution and segmentation of ultrasound images. PP-ULA employs the majorize-minimize principle to address non log-concave priors. As shown in numerical experiments, PP-ULA leads to a significant time reduction and to very satisfactory deconvolution and segmentation results on both simulated and real ultrasound data.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

Notation

$\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{R}^*$	: sets of real, positive real and non-zero real numbers
$\mathbb{R}^m$	: set of vectors with m entries
$\mathbb{R}^{m\times n}$	: set of matrices with m rows and n columns
$\mathbb{N}$, $\mathbb{N}^*$	: sets of positive and non-zero positive integers
$\mathfrak{S}_m$	: set of symmetric matrices in $\mathbb{R}^{m\times m}$
$\mathfrak{S}_m^+$	: set of symmetric positive-definite matrices in $\mathbb{R}^{m\times m}$
$\preceq$	: Loewner partial order for matrices
s	: scalars will be denoted by lowercase letters
$\boldsymbol{v}$	: vectors will be denoted by lowercase bold letters
$\boldsymbol{M}$	: matrices will be denoted by uppercase bold letters
$\boldsymbol{M}^\top$, $\boldsymbol{M}^{-1}$	: transpose and inverse of $\boldsymbol{M}$, respectively
$\boldsymbol{I}_m$	: square identity matrix in $\mathbb{R}^{m\times m}$
$\mathbf{1}_m$	: vector of $\mathbb{R}^m$ with all entries equal to 1
$\mathbf{0}_{m\times n}$	: matrix in $\mathbb{R}^{m\times n}$ with all entries equal to 0
$\mathbf{0}_m$	: vector of $\mathbb{R}^m$ with all entries equal to 0
v_i	: ith coefficient of $\boldsymbol{v}$
$M_{i,j}$	: element in the ith row and jth column of $\boldsymbol{M}$
$\lvert s\rvert$	: absolute value of s
$\langle\cdot,\cdot\rangle$	: Euclidean inner product
$\lVert\boldsymbol{v}\rVert$	: ℓ_2 norm of $\boldsymbol{v}$
$\lVert\boldsymbol{v}\rVert_{\boldsymbol{M}}$	: norm induced by $\boldsymbol{M}\in\mathfrak{S}_m^+$ and equal to $\langle\boldsymbol{v},\boldsymbol{M}\boldsymbol{v}\rangle^{\frac{1}{2}}$
$\lvert\!\lvert\!\lvert\boldsymbol{M}\rvert\!\rvert\!\rvert$	: spectral norm of $\boldsymbol{M}$
$\mathrm{Diag}(\boldsymbol{v})$	: diagonal matrix whose elements are given by $\boldsymbol{v}$
$\nabla f(\boldsymbol{v})$	: gradient vector of the function f at $\boldsymbol{v}$
$\nabla^2 f(\boldsymbol{v})$	: Hessian matrix of f at $\boldsymbol{v}$
$\imath_{\mathcal{C}}$	: indicator function of the set $\mathcal{C}$
$\mathrm{dist}(\boldsymbol{v},\mathcal{C})$	: $\inf_{\boldsymbol{w}\in\mathcal{C}}\lVert\boldsymbol{v}-\boldsymbol{w}\rVert$, Euclidean distance between $\boldsymbol{v}$ and the set $\mathcal{C}$
$\mathcal{N}(\mu,\sigma^2)$	: Gaussian distribution with mean $\mu\in\mathbb{R}$ and variance $\sigma^2\in\mathbb{R}_+^*$
$\mathcal{U}(a,b)$	: uniform distribution in $[a,b]$

Acronyms

ADMM : alternating direction method of multipliers
A-PDS : accelerated primal-dual splitting
CNN : convolutional neural network
CNR : contrast-to-noise ratio
CT : computed tomography
DFB : dual forward-backward
DNN : deep neural network
FB : forward-backward
GFBS : generalized forward-backward splitting
HMC : Hamiltonian Monte Carlo
IPM : interior point method
KL : Kurdyka-Łojasiewicz
l.s.c. : lower semicontinuous
MAP : maximum a posteriori
MCMC : Markov Chain Monte Carlo
MH : Metropolis-Hasting
MM : majorize-minimize
MMSE : minimum mean squared error
OA : overall accuracy
PDS : primal-dual splitting
PSF : point-spread function
PSNR : peak signal-to-noise ratio
P-ULA : proximal unadjusted Langevin algorithm
ReLU : rectified linear unit
RF : radio-frequency
SSIM : structural similarity measure
TRF : tissue reflectivity function
US : ultrasound
VMFB : forward-backward variable metric

Contents

- Chapter 1 -

General introduction

§ 1.1 Context

In this work, we develop and investigate resolution methods for inverse problems, and more specifically for applications related to signal and image processing. This field has undergone major changes due to the emergence of very powerful deep learning methods, which compete with more traditional approaches, such as variational strategies or Bayesian methods.

At the crossroads of the aforementioned approaches lies optimization. Optimization algorithms occur either in the design or implementation of the methods used to solve inverse problems. Consequently, there is a need for reliable and adequate iterative solvers. In particular, dealing with the ever-growing size of databases and datapoints, for instance in 3D applications, requires powerful mathematical tools. The proximity operator notably plays an important role in large-scale and nondifferentiable optimization. It has led to competitive algorithms with convergence guarantees, available convergence rates and noteworthy practical performance. The goal of this thesis is to illustrate the versatility of proximal algorithms in the context of inverse problems. To that end, we develop three methods combining the proximity operator with different techniques.

Variational methods are classical model-based methods with well-known mathematical properties. The variational strategy consists in designing an objective function whose minimizer is an appropriate estimate of the desired signal or image. Hence, prior knowledge about the solution can be incorporated into the resulting optimization problem. This applies to interior point methods, which are well suited for constrained problems in linear programming. In Chapter 3, we show that the scope of the logarithmic barrier method can be extended by combining it with a proximal iterative scheme.

The generalization of data collection and the creation of diverse and large databases have enabled deep learning methods to reach state-of-the-

art results for tasks such as image restoration, compressive sensing, image recovery and classification, to name only a few. Together with the massive development of these approaches, concerns regarding their reliability and stability are growing. Neural networks include, by nature, many degrees of freedom, which makes their study very intricate and their interpretation limited. Although variational and deep learning methods are based on distinct principles, they have complementary benefits and drawbacks, and therefore can greatly benefit from each other. Accordingly, in Chapter 4 we propose an interpretable neural network architecture inspired from a proximal interior point algorithm.

Similarly to machine learning, stochastic simulation methods estimate automatically the variables of interest. The Bayesian framework requires knowledge of prior distributions for the involved variables. Therefore, like in variational strategies, desirable properties for the solution can be promoted based on the choice of these priors. Stochastic simulation methods also offer convergence guarantees. Nonetheless, they may suffer from prohibitively long computational times. In Chapter 5, we propose to accelerate a stochastic sampling method by introducing a novel update strategy based on proximal tools.

We show in numerical experiments that the proposed methods compare favorably with state-of-the-art approaches on several challenging image processing applications. Hence, the outcomes of our work have both a mathematical and an applicative component.

§ 1.2 Main contributions

Chapter 3 includes the following contributions.

(i) We combine the proximity operator with the logarithmic barrier method in a new proximal interior point algorithm called PIPA. One interesting feature of PIPA is its ability to handle variable metrics, which can be used to boost its convergence.

(ii) We also provide convergence results for this algorithm under a mild boundedness condition on the involved variable metrics. Furthermore, we show that the inner loop in PIPA has a linear convergence rate under some suitable assumptions.

(iii) Finally, the performance of the proposed algorithm is evaluated on two applications, namely hyperspectral unmixing, and joint geometry-texture decomposition and reconstruction of computed tomography data. PIPA compares favorably with state-of-the-art algorithms on these two examples.

Our main contributions from Chapter 4 are listed hereafter.

(i) We propose a novel neural network architecture called iRestNet, which is obtained by unfolding a proximal interior point algorithm over a finite number of iterations. One key feature of this network is that prior knowledge can be directly incorporated into iRestNet thanks to a logarithmic barrier.

(ii) We derive expressions for the proximity operator of the barrier and for its first derivatives for three useful types of contraints. In particular, this allows our network to be trained using backpropagation and gradient descent.

(iii) We study the robustness of the proposed network with regards to an input perturbation, and derive conditions ensuring its stability in a useful case.

(iv) Numerical experiments performed on several non-blind image deblurring examples show that iRestNet compares favorably with respect to other variational and deep learning methods, including other neural networks based on deep unfolding. Finally, iRestNet benefits from a short execution time per image without any parameter search.

In Chapter 5, we consider a Bayesian framework and contribute to this area as follows.

(i) We propose a new sampling algorithm referred to as PP-ULA. This method is an original accelerated preconditioned version of the proximal unadjusted Langevin algorithm. PP-ULA takes advantage of a splitting strategy. It also includes variable metrics used for acceleration.

(ii) We present an efficient solver based on the majorize-minimize principle to tackle the involved non log-concave priors.

(iii) The proposed PP-ULA is then incorporated within a hybrid Gibbs sampler, yielding a substantial reduction of the computational time needed to perform joint high-quality deconvolution and segmentation of ultrasound images.

§ 1.3 Collaborations

Some parts of this thesis have been done in collaboration with other researchers.

We have worked conjointly with Denis Kouamé and Jean-Yves Tourneret from the IRIT laboratory (UMR CNRS 5505), Toulouse, France, on a Bayesian approach for solving a challenging inverse problem in ultrasound imaging.

More precisely, the topic of this project was to accelerate a hybrid Gibbs sampler using proximal tools, with the aim of performing joint deconvolution and segmentation of ultrasound medical images. I spent two weeks in the TéSA laboratory in Toulouse as part of this collaboration. The results of this work are presented in Chapter 5.

We have also worked with Carla Bertocchi and Marco Prato from the university of Modena, Italy, on an optimization-inspired neural network architecture. Our goal was to apply deep unfolding to a proximal interior point algorithm. During this research collaboration, I spent one month in the university of Modena to work on this project. The corresponding results are presented in Chapter 4.

§ 1.4 Publications

For articles with the * symbol, authors are listed in the alphabetical order, as it is customary in mathematical journals.

Submitted journal articles

E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet (2019). A proximal interior point algorithm with applications to image processing*. In *hal-02120005*.

Accepted or published journal articles

C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, M. Prato (2018). Deep unfolding of a proximal interior point method for image restoration*. *To appear in Inverse Problems*, doi:10.1088/1361-6420/ab460a.

M.-C. Corbineau, D. Kouamé, E. Chouzenoux, J.-Y. Tourneret, J.-C. Pesquet (2019). Preconditioned P-ULA for joint deconvolution-segmentation of ultrasound images. In *IEEE Signal Processing Letters*, 26(10), pp.1456-1460, doi:10.1109/LSP.2019.2935610.

Conference proceedings

M.-C. Corbineau, C. Bertocchi, E. Chouzenoux, M. prato, J.-C. Pesquet. Learned image deblurring by unfolding a proximal interior point algorithm. In *Proceedings of the 26th IEEE International Conference on Image processing (ICIP)*. Taipei, Taïwan, September 2019.

M.-C. Corbineau, E. Chouzenoux, J.-C. Pesquet. Geometry-texture decomposition/reconstruction using a proximal interior point algorithm. In

Proceedings of the 10th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM). Sheffield, UK, July 2018, pp.435-439.

M.-C. Corbineau, E. Chouzenoux, J.-C. Pesquet. PIPA : a new proximal interior point algorithm for large-scale convex optimization. In *Proceedings of the 43rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada, April 2018, pp.1343-1347.

Talks

Deep unfolding of a proximal interior point algorithm for image restoration. In *mini-symposium 'MS-24 From inverse problems to machine learning and back'*, Applied Inverse Problems Conference (AIP). Grenoble, France, 8 July 2019.

Joint geometry-texture decomposition and reconstruction of CT scans using a proximal interior point algorithm. In *Young Researchers in Imaging Seminars*. Institut Henri Poincaré, Paris, France, 27 February 2019.

Proximal interior point algorithm for large scale image processing problems. In *mini-Symposium 'MS-59 Approaches for fast optimisation in imaging and inverse problems'*, SIAM Conference on Imaging Science. Bologna, Italy, 7 June 2018.

A proximal interior point algorithm for large-scale convex optimization. In *Journées annuelles 2017 des GdR 3273 mathématiques de l'optimisation et applications (MOA) et 2286 mathématiques de l'imagerie et de ses applications (MIA)*. Bordeaux, France, 18 Octobre 2017.

§ 1.5 Outline

This manuscript is organized as follows.

In Chapter 2, we provide a general formulation for the study of inverse problems in Section 2.1.1 and describe briefly the three resolution methods mentioned previously, namely variational strategies (Section 2.1.2), Bayesian methods (Section 2.1.3) and deep learning techniques (Section 2.1.4). In particular, we introduce in Section 2.1.4.3 the concept of deep unfolding used in machine learning. Then, we introduce in Section 2.2.1 the main mathematical tools and notation that are used in this thesis. We end this chapter by presenting the optimization algorithms used in the proposed methods or as comparisons in numerical experiments (Section 2.2.2-2.2.5). A special attention is devoted to interior point methods in Section 2.2.5.

Chapter 3 is dedicated to the study of a new proximal interior point algorithm. For clarity, after describing the proposed method in Section 3.3,

we provide our main convergence results in Section 3.4 before detailing the corresponding proofs in Section 3.5. Finally, the results from two numerical experiments are presented in Sections 3.6 and 3.7. The first application is hyperspectral unmixing, while in the second application two tasks are jointly performed: geometry-texture image decomposition, and reconstruction of computed tomography data.

In Chapter 4, we investigate a novel neural network architecture designed by unfolding a proximal interior point algorithm. We start by describing in Section 4.2 the algorithm which is at the core of our method. Then, in Section 4.3 we derive the expressions of the proximity operator of the barrier and of its first derivatives for three types of constraints. These results are followed by a description of the proposed architecture called iRestNet (Section 4.4), whose robustness to an input perturbation is studied in Section 4.5. Finally, in Section 4.6 we evaluate iRestNet and compare it with state-of-the-art methods in numerical experiments related to non-blind image deblurring.

In Chapter 5, we consider a Bayesian setting and study a new proximal sampling strategy with applications in ultrasound imaging. We first describe the considered inverse problem in Section 5.1, which can be formulated as a joint deconvolution and segmentation of ultrasound images. Next, we detail in Section 5.3 the proposed preconditioned proximal unadjusted Langevin algorithm, referred to as PP-ULA. We then present the investigated hierarchical Bayesian model in Section 5.4, whereupon the proposed hybrid Gibbs sampler including PP-ULA is detailed in Section 5.5. Lastly, numerical experiments on both simulated and real ultrasound data are presented in Section 5.6.

Finally, we draw some conclusions and perspectives in Chapter 6.

§ 1.6 General notation

Throughout this document, $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}^*$ denote the sets of real, positive real and non-zero real numbers, respectively. The set of vectors with m entries is noted $\mathbb{R}^m$, while the set of matrices with m rows and n columns is referred to as $\mathbb{R}^{m\times n}$. We use the standard notations $\mathbb{N}$ and $\mathbb{N}^*$ for the sets of positive and non-zero positive integers. Scalars will be denoted by lowercase letters, vectors will be denoted by lowercase bold letters, matrices will be denoted by uppercase bold letters and, finally, function will be denoted by lowercase or uppercase letters. For every vector $\boldsymbol{v} = (v_i)_{1\leq i\leq m} \in \mathbb{R}^m$, v_i denoted its ith coefficient. For every matrix $\boldsymbol{M} = (M_{i,j})_{1\leq i\leq m, 1\leq j\leq n} \in \mathbb{R}^{m\times n}$, $M_{i,j}$ denotes the element in the ith row and jth column of $\boldsymbol{M}$. For every matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, $\boldsymbol{M}^\top$, $\boldsymbol{M}^{-1}$ and $|||\boldsymbol{M}|||$ denote its transpose, inverse and spectral norm, respectively. The square identity matrix in $\mathbb{R}^{m\times m}$ is referred to as $\boldsymbol{I}_m$, while the vector of $\mathbb{R}^m$ with all entries equal to 1 is noted

$\mathbf{1}_m$. The matrix in $\mathbb{R}^{m\times n}$ with all entries equal to 0 is noted $\mathbf{0}_{m\times n}$, and the vector of $\mathbb{R}^m$ with all entries equal to 0 is denoted as $\mathbf{0}_m$. For every $s \in \mathbb{R}$, the absolute value of s is noted $|s|$. The Euclidean inner product is denoted as $\langle \cdot, \cdot \rangle$ and, for every $\boldsymbol{v} \in \mathbb{R}^m$, $\|\boldsymbol{v}\|$ is the ℓ_2 norm of $\boldsymbol{v}$.

- Chapter 2 -

Background

In many practical cases, the acquired data are an indirect measurement of the desired information. This can be due to physical constraints, for instance in astronomy, or to safety measures, as in medical diagnostics. Furthermore, the data are often degraded or incomplete, either because of the acquisition process or on account of restricted storage capacity. These limitations can induce a significant loss of information. In this context, estimating the sought signal or image constitutes a challenging *inverse problem*. Optimization plays a critical role in designing efficient algorithms and methods for solving problems of this type.

In this chapter, we provide a mathematical formulation for inverse problems in Section 2.1.1 and introduce three different resolution approaches that are investigated in this thesis: variational strategies (Section 2.1.2), stochastic simulations (Section 2.1.3) and deep learning (Section 2.1.4). Finally, after presenting some key mathematical tools in Section 2.2.1, we present in Sections 2.2.2-2.2.5 useful optimization algorithms that are used throughout this work. A special attention is devoted to interior point methods in Section 2.2.5.

§ 2.1 Inverse problems

Inverse problems arise in a wide range of domains such as astronomy [Lucy, 1994; Bertero et al., 2009], machine learning [De Vito et al., 2005; Jenatton et al., 2010], geophysics [Menke, 2018], image processing, which includes for instance image restoration [Benvenuto et al., 2008] and image super-resolution [Yang et al., 2010]; and medical imaging [Bertero and Piana, 2006], with applications like computed tomography (CT), magnetic resonance imaging, and fluorescence microscopy.

2.1.1 Formulation

In this thesis we consider the following formulation for inverse problems,

$$\boldsymbol{y} = \mathcal{D}(\mathcal{H}(\overline{\boldsymbol{x}})), \tag{2.1}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ gathers observations, $\overline{\boldsymbol{x}} \in \mathbb{R}^m$ is the variable of interest, $\mathcal{H} : \mathbb{R}^m \to \mathbb{R}^n$ is the *observation operator*, and $\mathcal{D} : \mathbb{R}^n \to \mathbb{R}^n$ is the *degradation operator*.

The degradation operator represents a loss of information, which can come from the devices used to collect observations. For instance, optical sensors are generally subject to internal fluctuations, which are usually modelled as an additive white Gaussian noise independent from the signal of interest [Oliveira et al., 2009]. There exist more realistic models where, for instance, the noise variance is a function of the sought signal [Moser, 2012]. In some applications such as medical ultrasound [Sudha et al., 2009] or synthetic aperture radar [López-Martínez and Fabregas, 2003], measurements are corrupted by speckle noise, which is a multiplicative noise [Durand et al., 2010]. We can also mention impulsive noise [Cai et al., 2010], Poisson noise [Salmon et al., 2014], which occurs in photon-counting devices, and noises that are a mixture of the abovementioned noise models [Chouzenoux et al., 2015].

Regarding the observation operator, it accounts for the fact that the observation is an indirect measurement of the desired signal. In image deblurring for instance, $\mathcal{H}$ is frequently modelled as a linear operator representing the circular convolution of the image with a blur caused by optical defects [Bertero et al., 2009] or sensor motion [Levin et al., 2009]. This operator is also linear in applications such as hyperspectral unmixing [Iordache et al., 2012] and compressive sensing [Dai and Milenkovic, 2009], but there exist nonlinear observation models too, e.g. [Dobigeon et al., 2013]. Throughout this thesis, we assume that $\mathcal{H}$ is known, i.e. we do not address blind inverse problems.

Solving problem (2.1) consists in finding a proper estimate $\boldsymbol{x}^*$ of $\overline{\boldsymbol{x}}$ given $\boldsymbol{y}$. Assume that the observation operator is a linear operator, modelled by a matrix $\boldsymbol{H} \in \mathbb{R}^{n \times m}$, and that the noise is Gaussian. Then, an estimate of $\overline{\boldsymbol{x}}$ can be produced by solving the following least-squares problem,

$$\underset{\boldsymbol{x} \in \mathbb{R}^m}{\text{minimize}} \ \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2, \tag{2.2}$$

where $\|\cdot\|$ is the Euclidean norm. However, (2.2) is generally ill-posed [Hadamard, 1902], which means that, sometimes, there does not exist a solution, or when it exists, this solution is not necessarily unique, and it can be very sensitive to the presence of noise. Hence, there is a need for more performant approaches for solving inverse problems.

In the next sections, we introduce three different strategies that are investigated in this thesis, namely variational strategies, stochastic simulation methods and neural networks.

2.1.2 Variational strategy

Inverse problems can be efficiently solved by finding a minimizer of a given objective function. We refer to this approach as *variational* strategy.

2.1.2.1 Formulation

In order to find an appropriate solution to an inverse problem like (2.1), variational methods incorporate prior information on the sought signal through a penalization term. This leads to the following minimization problem,

$$\underset{\boldsymbol{x}\in\mathcal{C}}{\text{minimize}} \;\; f(\mathcal{H}(\boldsymbol{x}), \boldsymbol{y}) + \lambda\mathcal{R}(\boldsymbol{x}), \tag{2.3}$$

where $f : \mathbb{R}^n \times \mathbb{R}^n \to]-\infty, +\infty]$ is a *data fidelity* function, $\mathcal{R} : \mathbb{R}^m \to]-\infty, +\infty]$ is a *regularization* function, $\lambda \in \mathbb{R}_+$ is a regularization parameter and $\mathcal{C}$ is a subset of $\mathbb{R}^m$. The data fidelity function is directly related to the degradation model $\mathcal{D}$. For instance, when the noise is assumed to be Gaussian with variance σ^2, then $f(\cdot, \boldsymbol{y}) = \frac{1}{2\sigma^2}\|\boldsymbol{y} - \cdot\|^2$. The regularization term allows to enforce some desirable property in the solution like sparsity [Tibshirani et al., 2005; Zhu et al., 2015], smoothness [Poddar and Jacob, 2015] or grouping [Zou and Hastie, 2005], while the feasible set $\mathcal{C}$ can embed problem-related constraints [Musse et al., 2001; Klodt and Cremers, 2011]. Some classical choices for $\mathcal{R}$ are sparsity-promoting functions [Bouman and Sauer, 1996; Bect et al., 2004; Bach et al., 2012; Pustelnik et al., 2016] and the total variation and its various extensions [Rudin et al., 1992; Chambolle, 2004; Aujol, 2009], which promote piecewise constant solutions. The regularization parameter λ is used to weight the prior information with respect to the data fidelity term. When problem (2.3) does not have a closed-form solution, it must be solved by an iterative solver such as the ones presented in Section 2.2.

2.1.2.2 Limitations

Although useful, variational approaches are sometimes limited by their complexity: solving (2.3) may require advanced algorithms that may be too slow for real-time applications. In addition, λ is a parameter that needs to be set and $\mathcal{R}$ is usually parametrized by one or several parameters, whose optimal choice may strongly depend on the data at hand. These parameters are often tuned manually or computed using, for instance, cross validation, the discrepancy principle [Scherzer, 1993], or methods based on Stein unbiased risk estimates (SURE) [Deledalle et al., 2014]. However, these methods are often time-consuming and their success is not always guaranteed. Furthermore, despite numerous efforts in designing sophisticated models, the solution to (2.3) could be further away from $\overline{\boldsymbol{x}}$ than an intermediate iterate produced by

a given algorithm used for solving (2.3). Such phenomenon justifies the development of early stopping methods, where the iterative procedure is stopped before convergence [Yao et al., 2007; Rosasco and Villa, 2015]. Finding the optimal stopping time depends on the algorithm and requires the use of an oracle such as SURE, which may explain why these techniques are currently restricted to relatively simple objective functions.

In view of these limitations, an alternative to classical variational methods is the Bayesian approach. Accordingly, we present in the next section stochastic simulation methods, which can be used to solve inverse problems in a Bayesian setting.

2.1.3 Bayesian approaches

Bayesian theory leads to useful simulation methods that have demonstrated their efficiency for solving inverse problems such as image restoration [Besag et al., 1991], classification [Corander et al., 2006], compressive sensing [Xu et al., 2014b] and source separation [Moussaoui et al., 2006]. In this section we consider a Bayesian setting, in the sense that $\boldsymbol{y}$ and $\boldsymbol{x}$ from Section 2.1.1 are assumed to be random variables [Bernardo and Smith, 2009]. After highlighting the links between Bayesian approaches and variational strategies, we are going to briefly introduce the simulation methods used in Chapter 5.

2.1.3.1 Links with the variational strategy

Although stochastic simulation approaches are quite different from variational methods, Bayesian theory provides an insightful reinterpretation of the data fidelity term and regularization function introduced in Section 2.1.2. Let $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})$ be the likelihood of the observations and $\mathrm{p}(\boldsymbol{x})$ be the prior density associated to the variable of interest. Assume that $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}) \propto \exp(-f(\boldsymbol{x}, \boldsymbol{y}))$ and that $\mathrm{p}(\boldsymbol{x}) \propto \exp(-\mathcal{R}(\boldsymbol{x}))$. Then, problem (2.3) is equivalent to

$$\underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{maximize}} \;\; \mathrm{p}(\boldsymbol{y}|\boldsymbol{x})\mathrm{p}(\boldsymbol{x}). \tag{2.4}$$

According to the Bayes rules we have $\mathrm{p}(\boldsymbol{y}|\boldsymbol{x})\mathrm{p}(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{y})\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$, where $\mathrm{p}(\boldsymbol{y})$ is the prior density of the observations, and $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ is the posterior density of the sought variable. Hence, problem (2.4) can be reformulated as

$$\underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{maximize}} \;\; \mathrm{p}(\boldsymbol{x}|\boldsymbol{y}), \tag{2.5}$$

and a solution given by the variational strategy from Section 2.1.2 is then the maximum a posteriori (MAP) estimator of $\overline{\boldsymbol{x}}$ [Lermé et al., 2014].

It is worth noting that Bayesian approaches can be used to obtain other estimators than the MAP, such as the minimum mean squared error (MMSE) estimator [Lesieur et al., 2015], i.e. the expectation of $\boldsymbol{x}$ with regards to the density $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$. In addition, the Bayesian setting can help

derive useful uncertainty information about the estimate, for instance its variance.

In the next section, we introduce a class of stochastic methods called Markov chain Monte Carlo methods, which are widely used for solving inverse problems.

2.1.3.2 Markov chain Monte Carlo methods

Considering an inverse problem as defined in Section 2.1, the goal of Bayesian methods is to estimate $\overline{\boldsymbol{x}}$ from the posterior probability distribution $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$. When this law is not easy to sample from, one can rely on Markov chain Monte Carlo (MCMC) methods [Brooks et al., 2011; Robert and Casella, 2013; Pereyra et al., 2016], which generate a Markov chain whose stationary distribution is the target law. In Chapter 5 we use two well-known MCMC methods, which are the Metropolis-Hastings random walk algorithm and the Gibbs sampler.

Metropolis-Hastings random walk algorithm

The Metropolis-Hastings (MH) random walk algorithm proposed in [Roberts et al., 1997] is a special case of the original MH sampling method from [Metropolis et al., 1953] and [Hastings, 1970]. For every $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\sigma \in \mathbb{R}_+^*$ we denote by $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I}_m)$ the Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\sigma^2 \boldsymbol{I}_m$. In addition, for every $(a, b) \in \mathbb{R}^2$ such that $a \leq b$, we refer to the uniform distribution in $[a, b]$ as $\mathcal{U}(a, b)$. Algorithm 1 draws sample from the target distribution $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ by generating candidates from a proposal law that are accepted with a certain probability. These candidate samples are produced by adding to the current iterate a perturbation which is proportional to a realization of the standard normal distribution. Then, an acceptance test is performed such that, samples that are more probable to be produced by the target law than the current iterate are always accepted, and samples that are less probable states are accepted with a probability $\mathrm{p}(\tilde{\boldsymbol{x}}_{k+1}|\boldsymbol{y})/\mathrm{p}(\boldsymbol{x}_k|\boldsymbol{y}) < 1$.

The scale γ of the perturbation is a key element that drives the speed of convergence of Algorithm 1. In practice, the first iterations of the random walk are used to adjust γ based on some heuristics derived from the acceptance rate [Gelman et al., 1996]. The iterates produced during this *burn-in* period are then discarded from the Markov chain. Therefore, the computation of the empirical MMSE estimate $\boldsymbol{x}_{\mathrm{MMSE}}$ from T samples $(\boldsymbol{x}^{(t_0+t)})_{1 \leq t \leq T}$ produced after t_0 burn-in iterations reads

$$\boldsymbol{x}_{\mathrm{MMSE}} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}^{(t_0+t)}. \tag{2.6}$$

Algorithm 1: Metropolis-Hastings random walk [Roberts et al., 1997]

Let $\gamma > 0$.
Initialization: $\boldsymbol{x}_0 \in \mathbb{R}^m$.
for $k = 0, 1, \ldots$ **do**
  $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_k + \gamma \boldsymbol{\omega}_k$ with $\boldsymbol{\omega}_k \sim \mathcal{N}(\mathbf{0}_m, \boldsymbol{I}_m)$;
  $t_k \sim \mathcal{U}(0, 1)$;
  if $t_k < \min\left(1, \frac{p(\tilde{\boldsymbol{x}}_{k+1}|\boldsymbol{y})}{p(\boldsymbol{x}_k|\boldsymbol{y})}\right)$ **then**
    Accept: $\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{x}}_{k+1}$;
  else
    Reject: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$;
  end
end

Gibbs sampler

One can take advantage of the specific structure of a variable $\boldsymbol{x} \in \mathbb{R}^m$ that can be decomposed into $q \in \mathbb{N}$ blocks $(\boldsymbol{x}^{(i)})_{1 \leq i \leq q}$, such that $(\forall i \in \{1, \ldots, q\})$ $\boldsymbol{x}^{(i)} \in \mathbb{R}^{m_i}$ and $\sum_{i=1}^{q} m_i = m$. The Gibbs sampler [Casella and George, 1992], which is detailed in Algorithm 2, consists in sampling alternatively each block from its conditional distribution given observations $\boldsymbol{y}$ and the latest samples for the other blocks. This strategy is relevant when sampling from the individual conditional densities is simpler than drawing samples from the joint distribution. It can be shown that, after a long enough burn-in period, the samples generated by the Gibbs sampler follow the target law $\mathrm{p}(\boldsymbol{x}|\boldsymbol{y})$ [Robert and Casella, 2013, Section 10.2].

Algorithm 2: Gibbs sampler

Initialization: $\boldsymbol{x}_0 = (\boldsymbol{x}^{(i)})_{1 \leq i \leq q} \in \mathbb{R}^m$. **for** $k = 0, 1, \ldots$ **do**
  Generate $\boldsymbol{x}_{k+1}^{(1)} \sim \mathrm{p}\left(\boldsymbol{x}^{(1)}|\boldsymbol{y}, \boldsymbol{x}_k^{(2)}, \ldots, \boldsymbol{x}_k^{(q)}\right)$;
  Generate $\boldsymbol{x}_{k+1}^{(2)} \sim \mathrm{p}\left(\boldsymbol{x}^{(2)}|\boldsymbol{y}, \boldsymbol{x}_{k+1}^{(1)}, \boldsymbol{x}_k^{(3)}, \ldots, \boldsymbol{x}_k^{(q)}\right)$;
  $\vdots$
  Generate $\boldsymbol{x}_{k+1}^{(q)} \sim \mathrm{p}\left(\boldsymbol{x}^{(q)}|\boldsymbol{y}, \boldsymbol{x}_{k+1}^{(1)}, \ldots, \boldsymbol{x}_{k+1}^{(q-1)}\right)$;
end

When Algorithm 2 is combined with other MCMC methods, it is called a *hybrid Gibbs sampler*. For instance, when a MH step is used to sample each block, then the convergence of the resulting hybrid Gibbs sampler still holds [Robert and Casella, 2013, Section 10.3].

Although complex inverse problems can be solved using the Bayesian framework, sampling methods can be of prohibitive computational cost. As mentioned previously, Markov chains require burn-in iterations before reaching the target distribution. Then, more iterations are needed to explore this distribution. Furthermore, the fact that the produced samples are not systematically accepted can slow down the process. These different factors can result in a very large computational cost for MCMC methods. Nonetheless, MCMC methods can sometimes be accelerated using specific techniques [Robert et al., 2018]. We address this topic in Chapter 5, where we propose an acceleration method related to the proximal sampling scheme from [Pereyra, 2016].

In the next section, we introduce deep learning methods, which are investigated in Chapter 4.

2.1.4 Neural networks for solving inverse problems

As detailed in the review article [McCann et al., 2017], neural networks provide a popular and efficient alternative to variational methods for solving inverse problems. We clarify in Section 2.1.4.1 the vocabulary which is going to be used regarding neural networks. Then, we discuss in Section 2.1.4.2 some of their characteristics in the context of inverse problems. Lastly, we present in Section 2.1.4.3 the concept of *deep-unfolding*, which is at the core of Chapter 4.

2.1.4.1 Scope and vocabulary

In Chapter 4, we will focus on *feedforward* neural networks [Schmidhuber, 2015]. The latters do not include any cycle, as opposed to *recurrent* neural networks. A neural network can be thought of as a function $h(\cdot, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^p$ gathers parameters that are usually trained in a *supervised* manner [Goodfellow et al., 2016, Section 5.1]. This means that a training set of observations $(\boldsymbol{y}_j)_{1 \leq j \leq n}$ with associated solutions $(\overline{\boldsymbol{x}}_j)_{1 \leq j \leq n}$ to problem (2.1) is available. In this context, parameters in $\boldsymbol{\theta}$ are trained such that the outputs $(h(\boldsymbol{y}_j, \boldsymbol{\theta}))_{1 \leq j \leq n}$ generated by the network on the training set minimize a given objective function. The latter is referred to as *loss function*, and is supposed to measure the distance between the solutions to the considered inverse problem and the outputs of the network. This training step may be accelerated by using *batch normalization* after each layer of the neural network [Ioffe and Szegedy, 2015]. A *validation set* can also help monitoring the training without explicitely acting on the trainable parameters of the network [Goodfellow et al., 2016, Section 5.3]. This thesis does not address issues related to *unsupervised* training.

Although this training step can be both time and resource consuming, it

is generally performed offline. Hence, as opposed to classical variational and MCMC methods, neural networks benefit from very competitive execution times. This has contributed to their considerable development, including for solving inverse problems.

For some inverse problems, it might be easier for the network to learn the *residual* $\boldsymbol{y} - \overline{\boldsymbol{x}}$ instead of the solution $\overline{\boldsymbol{x}}$ itself, because pushing the residual to zero may appear easier than fitting an identity mapping by a stack of layers [He et al., 2016; Zhang et al., 2017b]. This strategy is called *residual learning* and is implemented using *skip connections* between layers. This strategy may also be used to avoid the problem of vanishing gradient during training [Tong et al., 2017].

Simple feedforward networks can be expressed as

$$\mathcal{R}_{K-1} \circ (\boldsymbol{W}_{K-1} \cdot + \boldsymbol{b}_{K-1}) \circ \cdots \circ \mathcal{R}_0 \circ (\boldsymbol{W}_0 \cdot + \boldsymbol{b}_0), \tag{2.7}$$

where $K \in \mathbb{N}^*$ is the number of layers, $(\mathcal{R}_k)_{0 \leq k \leq K-1}$ are nonlinear activation functions, $(\boldsymbol{W}_k)_{0 \leq k \leq K-1}$ are weight operators, also called linear layers, and $(\boldsymbol{b}_k)_{0 \leq k \leq K-1}$ are bias parameters. Among classical activation functions, one can mention the rectified linear unit (ReLU), $x \mapsto \max(0, x)$, and Sigmoid, $x \mapsto 1/(1 + \exp(-x))$. In Chapter 4, we use two common linear layers that are the *fully connected* [Pal and Mitra, 1992] and *convolutional* layers [Goodfellow et al., 2016, Chapter 9]. In convolutional layers, the upsampling factor between the filter points is called *dilation factor*. Augmenting the dilation factor enables to increase the receptive field of the filter without increasing the number of its parameters [Holschneider et al., 1990; Chen et al., 2017].

2.1.4.2 Neural networks and inverse problems

Deep Neural Networks (DNNs), and in particular Convolutional Neural Networks (CNNs), have demonstrated outstanding performance for various applications, such as denoising [Zhang et al., 2017b], non-blind and blind deblurring [Xu et al., 2014a; Schuler et al., 2013, 2016], super-resolution [Ledig et al., 2017], or computed-tomography reconstruction [Jin et al., 2017]. As detailed in [McCann et al., 2017], DNNs for inverse problems are very often preceded by a pre-processing step, which consists in finding a rough estimation of the sought signal by solving (2.2). This procedure tends, however, to strongly amplify noise. Hence, in this context, DNNs are often used as denoisers and artifact-removers. However, since prior knowledge about its output can hardly be incorporated into a DNN, which in most of the cases is viewed as a black-box, the explainability and reliability of such methods could be questioned [Szegedy et al., 2013]. Furthermore, the pre-processing step, in itself, can include a penalty, thus amounting to solving a problem of the form (2.3), where the regularization weight strongly depends on the noise level, e.g. [Schuler et al., 2013; Boublil et al., 2015].

The aforementioned limitations can be overcome by using *deep unfolding*, which combines neural networks and classical optimization algorithms. This approach is used in Chapter 4 and presented hereafter.

2.1.4.3 Deep unfolding

Consider a model-based method relying on an iterative solver $\mathcal{A}$. At each iteration $k \in \mathbb{N}$, this algorithm depends on several parameters which can be concatenated in a vector $\boldsymbol{\theta}_k \in \mathbb{R}^p$. These hyperparameters are either derived from the model or related to the algorithm itself. Assuming that the convergence of $\mathcal{A}$ is proven, several iterates are produced as follows for $k = 0, 1, \ldots$

$$\boldsymbol{x}_{k+1} = \mathcal{A}(\boldsymbol{x}_k, \boldsymbol{\theta}_k),$$

until an appropriate stopping criterion is satisfied. *Unfolding* or *unrolling* this method consists in setting the number of iterations to $K \in \mathbb{N}^*$, and treating the different hyperparameters $(\boldsymbol{\theta}_k)_{0 \leq k \leq K-1}$ as learnable parameters of a neural network with K layers [Hershey et al., 2014]. This leads to the following feed-forward architecture,

$$\mathcal{A}\left(\cdot, \mathcal{L}_{K-1}^{(\boldsymbol{\theta})}(\cdot)\right) \circ \cdots \circ \mathcal{A}\left(\cdot, \mathcal{L}_{0}^{(\boldsymbol{\theta})}(\cdot)\right),$$

where every hidden layer $\mathcal{L}_k^{(\boldsymbol{\theta})}$ is used to infer $\boldsymbol{\theta}_k$ as follows, $\mathcal{L}_k^{(\boldsymbol{\theta})}(\boldsymbol{x}_k) = \boldsymbol{\theta}_k$.

Interestingly, the fact that this technique makes use of a limited number of layers can be viewed as an analogue of early stopping methods [Rosasco and Villa, 2015]. It is however worth mentioning that, in unfolded algorithms, the number of iterations (i.e., layers) is tuned during the off-line training step and is then fixed for all test images, which differs from early stopping strategies where the iteration number usually differs for each processed image.

Related works apply deep unfolding to probabilistic models, such as Markov random fields [Hershey et al., 2014] or topic models [Chien and Lee, 2018], and to different algorithms like primal-dual solvers [Wang et al., 2016] or the proximal gradient method [Mardani et al., 2017; Diamond et al., 2017]. Classic optimization algorithms can be unfolded to perform many different tasks in image processing. For instance, FISTA and ISTA can be unfolded to perform sparse coding [Gregor and LeCun, 2010; Kamilov and Mansour, 2016], while the same ISTA and ADMM can be unfolded for image compressive sensing [Zhang and Ghanem, 2018; Sun et al., 2016]. However, in the aforementioned works, some functions and operators are learned, which weakens the link between the resulting network and the original algorithm. Deep unfolding is also used to learn shrinkage functions, which can be viewed as proximity operators of sparsity-promoting functions [Schmidt and Roth, 2014; Sun and Xu, 2015], or to optimize hyperparameters in nonlinear reaction diffusion models [Chen and Pock, 2017].

It can be stressed that optimization plays an important role in the three approaches that have been introduced in Sections 2.1.2-2.1.4 for solving inverse problems. For this reason, we present hereafter the main optimization algorithms that are used in this thesis.

§ 2.2 Optimization algorithms

Here, we focus on convex optimization problems of the following type,

$$\underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}} \;\; f(\boldsymbol{x}) + g(\boldsymbol{x}) + h(\boldsymbol{L}\boldsymbol{x}), \tag{2.8}$$

where $f : \mathbb{R}^m \to]-\infty, +\infty]$, $g : \mathbb{R}^m \to]-\infty, +\infty]$, g is assumed to be differentiable, $\boldsymbol{L} \in \mathbb{R}^{n\times m}$ and $h : \mathbb{R}^n \to]-\infty, +\infty]$. The purpose of this section is to present the optimization algorithms that either play a role in the proposed methods, or are used as comparisons in the numerical experiments. First, we present in Section 2.2.1 some mathematical notions that are closely related to the algorithms presented in Sections 2.2.2-2.2.5.

2.2.1 Mathematical analysis tools

We start by introducing some mathematical definitions.

2.2.1.1 Notation and definitions

The following definitions set up our framework for convex analysis and subdifferential calculus.

Definition 2.2.1 Let $f : \mathbb{R}^m \to]-\infty, +\infty]$.

(i) The *domain* of f is the set defined by

$$\operatorname{dom}(f) = \{\boldsymbol{x} \in \mathbb{R}^m \mid f(\boldsymbol{x}) < +\infty\}.$$

(ii) The function f is *proper* if $\operatorname{dom}(f)$ is not empty.

(iii) The function f is *convex* if for every $\alpha \in]0, 1[$ the following holds,

$$(\forall (\boldsymbol{x}, \boldsymbol{y}) \in \operatorname{dom}(f)) \quad f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}).$$

(iv) The function f is *coercive* if

$$\lim_{\|\boldsymbol{x}\|\to+\infty} f(\boldsymbol{x}) = +\infty.$$

(v) The function f is *lower semicontinuous* (l.s.c.) if, for every $\boldsymbol{x}_0 \in \mathbb{R}^m$,

$$\liminf_{\boldsymbol{x} \to \boldsymbol{x}_0} f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0).$$

(vi) The *conjugate* of f is the function $f^* : \mathbb{R}^m \to [-\infty, +\infty]$ defined by

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad f^*(\boldsymbol{x}) = \sup_{\boldsymbol{y} \in \mathbb{R}^m} \left(\langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{y}) \right).$$

We denote by $\Gamma_0(\mathbb{R}^m)$ the set of functions from $\mathbb{R}^m$ to $]-\infty, +\infty]$ that are proper, l.s.c. and convex. We use the notion of subdifferential as it was introduced by Moreau [Moreau, 1966] and Rockafellar [Rockafellar, 1970] in the 1960's, and whose definition is recalled below.

Definition 2.2.2 Let $f : \mathbb{R}^m \to]-\infty, +\infty]$. The *subdifferential* of f is the set-valued operator ∂f defined by

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \partial f(\boldsymbol{x}) = \{\boldsymbol{u} \in \mathbb{R}^m \mid (\forall \boldsymbol{y} \in \mathbb{R}^m) \ \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{u} \rangle + f(\boldsymbol{x}) \leq f(\boldsymbol{y})\}.$$

The subdifferential can be linked to the set of minimizers of a function using Fermat's rule, as stated below.

Theorem 2.2.3 (Fermat's rule) *[Bauschke and Combettes, 2017, Theorem 16.3] Let $f \in \mathbb{R}^m \to]-\infty, +\infty]$ be proper. Then,*

$$\operatorname*{Argmin}_{\boldsymbol{x} \in \mathbb{R}^m} f(\boldsymbol{x}) = \{\boldsymbol{x} \in \mathbb{R}^m \mid \mathbf{0}_m \in \partial f(\boldsymbol{x})\}.$$

In the next section, we introduce a fundamental tool in nonconvex optimization.

2.2.1.2 Kurdyka-Łojasiewicz property

The Kurdyka-Łojasiewicz (KL) inequality was first introduced by Łojasiewicz for real analytic functions [Lojasiewicz, 1963], and then extended by Kurdyka [Kurdyka, 1998] to differentiable functions that are definable in a o-minimal structure [van den Dries et al., 1994]. Following the seminal work of Łojasiewicz and Kurdyka, the KL property has been extensively used for proving the convergence [Bolte et al., 2007, 2014; Chouzenoux et al., 2014b] and obtaining convergence rates [Attouch and Bolte, 2009; Attouch et al., 2010] of optimization methods for possibly nondifferentiable functions, both in the convex and nonconvex case.

The KL inequality is based on the idea that a large number of functions can be re-parametrized such that they are sharp around their minimizers. This re-parametrization is based on the class of functions specified below.

Definition 2.2.4 For every $\eta \in \mathbb{R}_+^*$, Φ_η is the set of continuous concave functions $\phi : [0, \eta[\to [0, +\infty[$ such that

(i) $\phi(0) = 0$,

(ii) ϕ is $\mathcal{C}^1$ on $]0, \eta[$ and continuous at 0,

(iii) for every $s \in]0, \eta[$, $\phi'(s) > 0$.

We introduce the following notation for the Euclidean distance between a vector $\boldsymbol{v} \in \mathbb{R}^m$ and a set $\mathcal{C}$, $\operatorname{dist}(\boldsymbol{v}, \mathcal{C}) = \inf_{\boldsymbol{w} \in \mathcal{C}} \|\boldsymbol{v} - \boldsymbol{w}\|$. A general form of the KL inequality has been proposed in [Bolte et al., 2014] and is recalled hereafter.

Definition 2.2.5 [Bolte et al., 2014, Definition 3] Let $f : \mathbb{R}^m \to]-\infty, +\infty]$ be proper and l.s.c. The function is said to have the *Kurdyka-Łojasiewicz* property at $\overline{\boldsymbol{x}} \in \operatorname{dom}(\partial f) = \{\boldsymbol{x} \in \mathbb{R}^m \mid \partial f(\boldsymbol{x}) \neq \varnothing\}$ if there exist $\eta \in \mathbb{R}_+^*$, a neighborhood Ω of $\overline{\boldsymbol{x}}$, and a function $\phi \in \Phi_\eta$ such that, for all $\boldsymbol{x}$ such that

$$f(\overline{\boldsymbol{x}}) < f(\boldsymbol{x}) < f(\overline{\boldsymbol{x}}) + \eta,$$

the following inequality holds

$$\phi'(f(\boldsymbol{x}) - f(\overline{\boldsymbol{x}}))\ \operatorname{dist}(\mathbf{0}_m, \partial f(\boldsymbol{x})) \geq 1.$$

If f satisfies the KL property at each point of $\operatorname{dom}(\partial f)$, then f is called a *KL function*.

The KL property is satisfied in most practical optimization applications and, for a wide class of functions, the re-parametrization ϕ has a specific form which is detailed below and illustrated in Figure 2.1.

Definition 2.2.6 [Li and Pong, 2018, Definition 2.3] Let $f : \mathbb{R}^m \to]-\infty, +\infty]$ be a proper l.s.c. function satisfying the KL property at $\overline{\boldsymbol{x}} \in \operatorname{dom}(\partial f)$ within a neighborhood Ω for $\eta \in \mathbb{R}_+^*$. If there exist $\overline{c} \in \mathbb{R}_+^*$ and $\alpha \in [0, 1[$ such that the corresponding re-parametrization $\phi \in \Phi_\eta$ can be chosen as

$$(\forall s \in [0, \eta[) \quad \phi(s) = \overline{c} s^{1-\alpha},$$

then f is said to have the KL property at $\overline{\boldsymbol{x}}$ with *exponent* α. In other words, there exists $c \in \mathbb{R}_+^*$ such that, for every $\boldsymbol{x} \in \Omega$ such that $f(\overline{\boldsymbol{x}}) < f(\boldsymbol{x}) < f(\overline{\boldsymbol{x}}) + \eta$, the following inequality is satisfied,

$$\operatorname{dist}(\mathbf{0}_m, \partial f(\boldsymbol{x})) \geq c(f(\boldsymbol{x}) - f(\overline{\boldsymbol{x}}))^\alpha.$$

If f has the same exponent α at any $\overline{\boldsymbol{x}} \in \operatorname{dom}(\partial f)$, then f is said to be a KL function with *exponent* α.

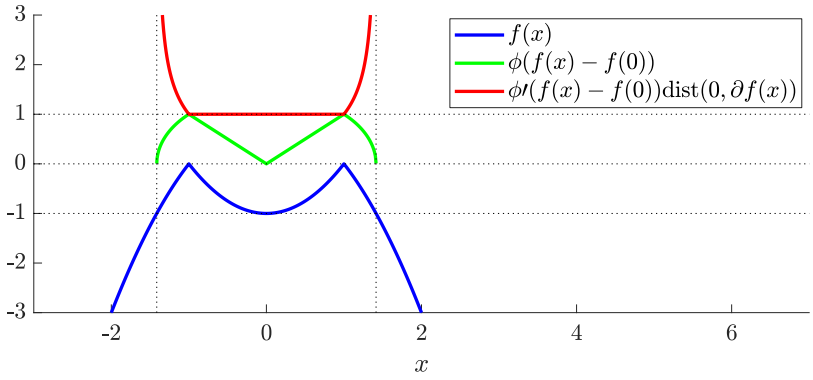


Figure 2.1: *Illustration of the KL re-parametrization of the nonconvex function* $f : x \in \mathbb{R} \mapsto -|x^2 - 1|$*, which is a KL function with exponent* $\frac{1}{2}$ *[Li and Pong, 2018, Corollary 5.2], with* $\phi : s \in [0, 1[\mapsto s^{\frac{1}{2}}$ *at the point* $\overline{x} = 0$*.*

It is worth noting that this particular form of ϕ encompasses the original inequality introduced by Łojasiewicz in the 1960's. In addition, the KL exponent can be computed explicitly under some conditions, as detailed in [Li and Pong, 2018]. This is of particular interest because this exponent can help to derive convergence rates for many optimization methods, as proven for instance in [Attouch and Bolte, 2009, Theorem 2] or [Attouch et al., 2010, Theorem 3.4].

In the next section we introduce the proximity operator, which is a very useful mathematical tool in optimization.

2.2.1.3 Proximity operators

The proximity operator has been first introduced in [Moreau, 1965] as an extension of the projection on a closed convex set. The proximity operator of a convex function $f \in \Gamma_0(\mathbb{R}^m)$ at a point $\boldsymbol{x} \in \mathbb{R}^m$ is then defined as the unique minimizer of $\boldsymbol{y} \mapsto f(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2$, namely,

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \mathrm{prox}_f(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \mathbb{R}^m}{\mathrm{argmin}}\, f(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2.$$

The above definition can be extended to nonconvex functions [Attouch and Bolte, 2009]. Furthermore, it can be made more general by using a weighted norm instead of the Euclidean norm [Becker and Fadili, 2012; Combettes and Vũ, 2014; Chouzenoux et al., 2014b]. In this thesis, weighted norms will be noted as follows; for every $\boldsymbol{v} \in \mathbb{R}^m$, $\|\boldsymbol{v}\|_{\boldsymbol{M}} = \langle \boldsymbol{v}, \boldsymbol{M}\boldsymbol{v} \rangle^{\frac{1}{2}}$ is the norm of $\boldsymbol{v}$ induced by $\boldsymbol{M} \in \mathfrak{S}_m^+$, with $\mathfrak{S}_m^+$ the set of symmetric positive-definite matrices in $\mathbb{R}^{m \times m}$.

These considerations lead to the following general definition, which will be used in the rest of this document.

Definition 2.2.7 Let $f : \mathbb{R}^m \to]-\infty, +\infty]$ be proper and l.s.c., and let $\boldsymbol{M} \in \mathfrak{S}_m^+$ and $\boldsymbol{x} \in \mathbb{R}^m$. The *proximity operator* of f at $\boldsymbol{x}$ with regards to the norm induced by $\boldsymbol{M}$ is defined as the set

$$\operatorname{prox}_f^{\boldsymbol{M}}(\boldsymbol{x}) = \operatorname*{Argmin}_{\boldsymbol{y} \in \mathbb{R}^m} \left(f(\boldsymbol{y}) + \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{x}\|_{\boldsymbol{M}}^2 \right).$$

If the above set is reduced to a singleton, then, for simplicity, this unique element will also be noted $\operatorname{prox}_f^{\boldsymbol{M}}(\boldsymbol{x})$. When the Euclidean norm is used, i.e. $\boldsymbol{M} = \boldsymbol{I}_m$, the proximity operator will be noted $\operatorname{prox}_f(\boldsymbol{x})$.

Throughout this thesis, $\boldsymbol{M}$ and its induced normed will be referred to as *preconditioner* and *variable metric*, respectively. It is worth noting that, if $f \in \Gamma_0(\mathbb{R}^m)$, then the proximity operator of f is uniquely defined on $\mathbb{R}^m$ since the objective function to be minimized is strongly convex.

Remark 2.2.8 Let $f \in \Gamma_0(\mathbb{R}^m)$, $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{M} \in \mathfrak{S}_m^+$. The proximity operator of f at $\boldsymbol{x}$ in the norm induced by $\boldsymbol{M}$ can be re-written as

$$\operatorname{prox}_f^{\boldsymbol{M}}(\boldsymbol{x}) = M^{-\frac{1}{2}} \operatorname{prox}_{f \circ \boldsymbol{M}^{-\frac{1}{2}}} \left(\boldsymbol{M}^{\frac{1}{2}} \boldsymbol{x} \right).$$

This reformulation can be used to compute the proximity operator of a function in a variable metric.

In view of its widespread use in applications related to signal and image processing, the proximity operator has been extensively studied, in particular in the convex case [Hiriart-Urruty and Lemaréchal, 1996; Chaux et al., 2007; Briceño-Arias and Combettes, 2009; Bauschke and Combettes, 2017]. We mention below some useful properties of the proximity operator that are used in this thesis. This results correspond to [Bauschke and Combettes, 2017, Theorem 14.3(ii), Proposition 12.28, Proposition 16.44, Proposition 24.8(v) and Corollary 24.15], respectively.

Theorem 2.2.9 (Moreau's decomposition) *Let $f \in \Gamma_0(\mathbb{R}^m)$, $\boldsymbol{x} \in \mathbb{R}^m$ and $\gamma \in \mathbb{R}_+^*$. Then,*

$$\boldsymbol{x} = \operatorname{prox}_{\gamma f}(\boldsymbol{x}) + \gamma \operatorname{prox}_{\frac{f^*}{\gamma}} \left(\frac{\boldsymbol{x}}{\gamma} \right).$$

Proposition 2.2.10 (Nonexpansiveness) *Let $f \in \Gamma_0(\mathbb{R}^m)$ and $\boldsymbol{M} \in \mathfrak{S}_m^+$. Then, $\operatorname{prox}_f^{\boldsymbol{M}}$ is nonexpansive with respect to $\|\cdot\|_{\boldsymbol{M}}$.*

Proposition 2.2.11 (Characterization) *Let $f \in \Gamma_0(\mathbb{R}^m)$, $\boldsymbol{M} \in \mathfrak{S}_m^+$ and $\boldsymbol{x} \in \mathbb{R}^m$. Then,*

$$\boldsymbol{y} = \operatorname{prox}_f^{\boldsymbol{M}}(\boldsymbol{x}) \iff \boldsymbol{x} - \boldsymbol{y} \in \boldsymbol{M}^{-1} \partial f(\boldsymbol{y}).$$

Proposition 2.2.12 (Translation) *Let $f \in \Gamma_0(\mathbb{R}^m)$, $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^m$, $\gamma \in \mathbb{R}_+^*$ and $\mu \in \mathbb{R}^*$. Set $g = f(\mu \cdot - z)$. Then,*

$$\operatorname{prox}_{\gamma g}(\boldsymbol{x}) = \frac{1}{\mu}(y + \operatorname{prox}_{\gamma\mu^2 f}(\mu\boldsymbol{x} - \boldsymbol{y}))$$

Proposition 2.2.13 (Composition) *Let $f \in \Gamma_0(\mathbb{R})$, $\boldsymbol{x} \in \mathbb{R}^m$ and $y \in \mathbb{R}^m \setminus \{\mathbf{0}_m\}$. Set $g = f(\langle \cdot, y \rangle)$. Then,*

$$\operatorname{prox}_g(\boldsymbol{x}) = \boldsymbol{x} + \frac{\operatorname{prox}_{\|\boldsymbol{y}\|^2 f}(\langle \boldsymbol{x}, \boldsymbol{y} \rangle) - \langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{y}\|^2}\boldsymbol{y}.$$

An explicit or closed-form expression of the proximity operator has been derived for many useful functions, e.g. see [1] for a collection of known proximity operators with associated codes. Hereafter, we list two examples that are relevant to this thesis.

Example 2.2.14 In Chapter 5, we consider functions of the type $\|\cdot\|_p^p$ for $p \in \mathbb{R}_+^*$. Since the proximity operator of $\|\cdot\|_p^p$ is separable with regards to its entries, we consider here functions of the form $|\cdot|^p$. When $p \geq 1$, $|\cdot|^p$ is convex and we can obtain an closed-form expression of the proximity operator for some specific values of p. The resulting operators are illustrated in Figure 2.2.

(i) If $p = 1$, we have the ℓ_1-norm, which is widely used in optimization to promote sparsity [Bach et al., 2012], and whose proximity operator is known as the *soft-thresholding* operator defined by

$$(\forall \gamma \in \mathbb{R}_+^*)(\forall x \in \mathbb{R}) \quad \operatorname{prox}_{\gamma|\cdot|}(x) = \operatorname{sign}(x)\max\{|x| - \gamma\}.$$

(ii) If $p \in \{\frac{4}{3}, \frac{3}{2}, 2, 3, 4\}$, then we can use the results from [Chaux et al., 2007, Examples 4.3 and 4.4] which are recalled below for every $\gamma \in \mathbb{R}_+^*$ and every $x \in \mathbb{R}$.

$$\operatorname{prox}_{\gamma|\cdot|^p}(x) = \begin{cases} x + \frac{4\gamma}{3\times 2^{\frac{1}{3}}}\left((\xi - x)^{\frac{1}{3}} - (\xi + x)^{\frac{1}{3}}\right) & \text{if } p = \frac{4}{3}; \\ \qquad \text{with } \xi = \left(x^2 + \frac{256}{729}\gamma^3\right)^{\frac{1}{2}} & \\ x + \frac{9}{8}\gamma^2\operatorname{sign}(x)\left(1 - \left(1 + \frac{16|x|}{9\gamma^2}\right)^{\frac{1}{2}}\right) & \text{if } p = \frac{3}{2}; \\ \frac{x}{2\gamma+1} & \text{if } p = 2; \\ \operatorname{sign}(x)\frac{(1+12\gamma|x|)^{\frac{1}{2}}-1}{6\gamma} & \text{if } p = 3; \\ \left(\frac{\xi+x}{8\gamma}\right)^{\frac{1}{3}} - \left(\frac{\xi-x}{8\gamma}\right)^{\frac{1}{3}} & \text{if } p = 4; \\ \qquad \text{with } \xi = \left(x^2 + \frac{1}{27\gamma}\right)^{\frac{1}{2}} & \end{cases}$$

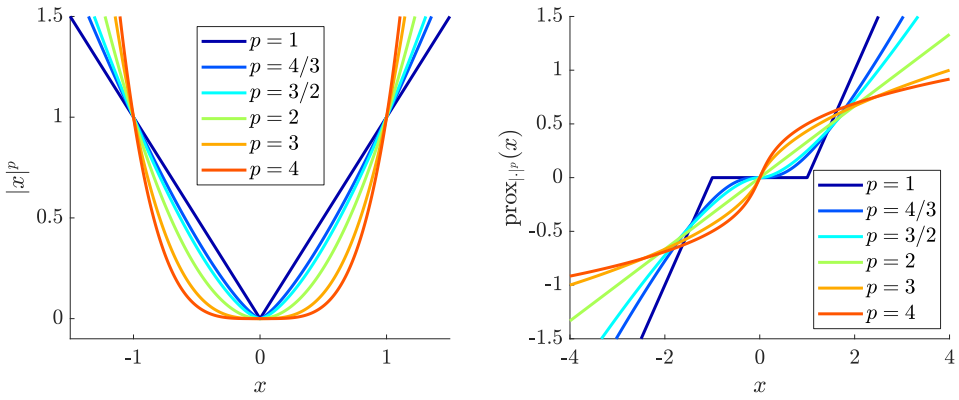


Figure 2.2: *Function $|\cdot|^p$ (left) and its proximity operator (right) for different values of p.*

Example 2.2.15 Our second example is the logarithmic barrier, which is classically used in interior point methods (see Section 2.2.5).

(i) The barrier associated with the feasible set $[0, +\infty[$, also called the negative Burg entropy, is the function defined by

$$(\forall x \in \mathbb{R}) \quad f(x) = \begin{cases} -\ln(x) & \text{if } x \in \mathbb{R}_+^*, \\ +\infty & \text{else.} \end{cases}$$

For every $\gamma \in \mathbb{R}_+^*$ we get the proximity operator of γf from [Bauschke and Combettes, 2017, Example 24.40] as follows,

$$(\forall x \in \mathbb{R}) \quad \mathrm{prox}_{\gamma f}(x) = \frac{x + (x^2 + 4\gamma)^{\frac{1}{2}}}{2}.$$

(ii) If we consider the compact feasible set $[a, b]$, with $a < b$, then the corresponding logarithmic barrier g is defined as follows,

$$(\forall x \in \mathbb{R}) \quad g(x) = \begin{cases} -\ln(x-a) - \ln(b-x) & \text{if } x \in]a, b[, \\ +\infty & \text{else.} \end{cases}$$

The proximity operator of g can be found in [Chaux et al., 2007, Example 4.15]: for every $\gamma \in \mathbb{R}_+^*$ and every $x \in \mathbb{R}$, $\mathrm{prox}_{\gamma g}(x) = \kappa$, where κ is the unique solution in $]a, b[$ to

$$\kappa^3 - (a + b + x)\,\kappa^2 + (ab + (a + b)\,x - 2\gamma)\,\kappa - abx + (a + b)\,\gamma = 0.$$

The solution κ to the above cubic equation can be found using Cardano's method. As one can see in Figure 2.3, the smaller γ is, the closer the proximity operator is to the projection on $[a, b]$.

In the next section, we introduce some properties related to operators. These notions are used in the convergence analysis presented in Chapter 3, and in the robustness study discussed in Section 4.5.

[1] http://proximity-operator.net/

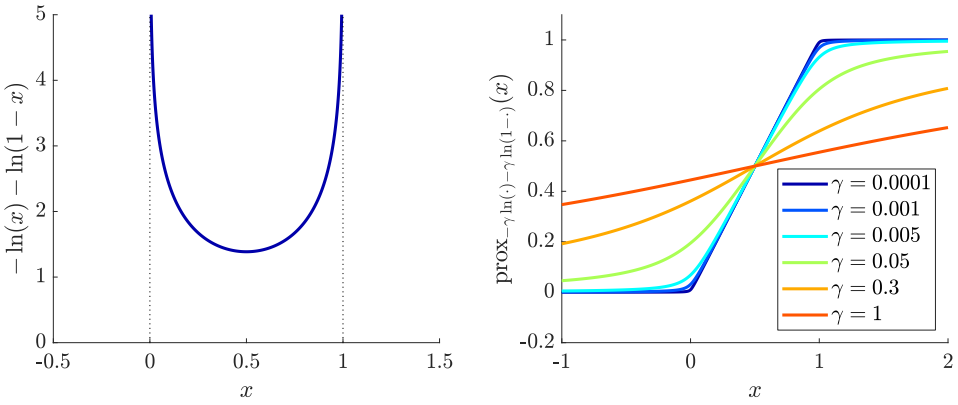


Figure 2.3: *Logarithmic barrier function (left) and its proximity operator (right) for different values of multiplicative factor γ.*

2.2.1.4 Operator properties

The Lipschitz continuity presented hereafter gives an upper bound for the rate of change of a given operator.

Definition 2.2.16 Let $T : \mathbb{R}^m \to \mathbb{R}^n$.

(i) The operator T is *Lipschitz continuous* with constant $\alpha \in \mathbb{R}_+$, or α-*Lipschitz continuous*, if

$$(\forall(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^m \times \mathbb{R}^m) \quad \|T(\boldsymbol{x}) - T(\boldsymbol{y})\| \leq \alpha\|\boldsymbol{x} - \boldsymbol{y}\|.$$

(ii) The operator T is *nonexpansive* if it is Lipschitz continuous with constant equal to 1.

(iii) A differentiable function $f : \mathbb{R}^m \to]-\infty, +\infty]$ is said to be *Lipschitz differentiable* with constant $\alpha \in \mathbb{R}_+$ if its gradient ∇f is α-Lipschitz continuous.

Next, we introduce the notion of averageness, which can be seen as a generalization of nonexpansiveness.

Definition 2.2.17 [Bauschke and Combettes, 2017, Definition 4.33] Let $T : \mathbb{R}^m \to \mathbb{R}^m$ and $\alpha \in]0, 1[$. The operator T is *averaged* with constant α, or α-*averaged*, if there exists a nonexpansive operator $R : \mathbb{R}^m \to \mathbb{R}^m$ such that $T = (1 - \alpha)\boldsymbol{I}_m + \alpha R$.

The following proposition provides an upper bound of the effect of an input perturbation, which depends on the averageness constant. In particular, the smaller the constant is, the more stable the operator is.

Proposition 2.2.18 *[Bauschke and Combettes, 2017, Remark 4.34, Proposition 4.35] Let $T : \mathbb{R}^m \to \mathbb{R}^m$ and $\alpha \in]0, 1]$.*

(i) If T is averaged, then it is nonexpansive.

(ii) The operator T is α-averaged if and only if, for every $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^m \times \mathbb{R}^m$,

$$\|T(\boldsymbol{x}) - T(\boldsymbol{y})\|^2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \frac{1-\alpha}{\alpha}\|(\boldsymbol{I}_m - T)(\boldsymbol{x}) - (\boldsymbol{I}_m - T)(\boldsymbol{y})\|^2.$$

Now that we have introduced important mathematical tools used in this thesis, we present in the next sections the main algorithms that we use either for designing the proposed methods or as comparisons.

2.2.2 Forward-backward algorithms

First, we consider the case when h is the zero function. Then, (2.8) reads

$$\underset{\boldsymbol{x} \in \mathbb{R}^m}{\text{minimize}} \;\; f(\boldsymbol{x}) + g(\boldsymbol{x}). \tag{2.9}$$

The well-known forward-backward (FB) algorithm [Chen and Rockafellar, 1997; Tseng, 2000] addresses separately the two terms in the composite objective function, which leads to two steps for each iteration, namely a gradient (i.e. forward) step on the differentiable term g, and a proximal (i.e. backward) step on the nonsmooth term f.

2.2.2.1 Variable metric forward-backward algorithm

The FB algorithm can be accelerated by using a variable metric. This idea was already introduced in the original version of the FB algorithm in [Chen and Rockafellar, 1997], where the authors obtain the convergence of the algorithm in the convex setting assuming that the preconditioning matrices converge to a given operator [Chen and Rockafellar, 1997, Theorem 5.2]. The variable metric forward-backward (VMFB) algorithm has recently been extended in [Combettes and Vũ, 2014], where the authors consider summable errors in the computation of the gradient and of the proximity operator. The convergence of this algorithm is obtained in [Combettes and Vũ, 2014, Theorem 4.1] under a monotonicity assumption on the variable metrics. Some recent works, relying on the KL inequality, extend this convergence result to nonconvex problems under milder boundedness conditions on the variable metrics [Chouzenoux et al., 2014b; Frankel et al., 2015].

Algorithm 3 corresponds to a simplified version of the VMFB algorithm without the summable errors. By taking $(\forall k \in \mathbb{N})$ $\boldsymbol{M}_k = \boldsymbol{I}_m$ in Algorithm 3, we retrieve the classical FB algorithm.

Algorithm 3: Variable metric forward–backward [Combettes and Vũ, 2014] (exact version)

For every $k \in \mathbb{N}$, let $\gamma_k \in \mathbb{R}_+^*$, $\lambda_k \in \mathbb{R}_+^*$, and $\boldsymbol{M}_k \in \mathfrak{S}_m^+$.
Initialization: Take $\boldsymbol{x}_0 \in \mathbb{R}^m$.
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k \left(\operatorname{prox}_{\gamma_k f}^{\boldsymbol{M}_k} \left(\boldsymbol{x}_k - \gamma_k \boldsymbol{M}_k^{-1} \nabla g(\boldsymbol{x}_k) \right) - \boldsymbol{x}_k \right)$;
end

2.2.2.2 Generalized forward-backward algorithm

The FB algorithm can be generalized to the case when f can be written as $f = \sum_{i=1}^p f_i$. Then, problem (2.8) reads

$$\underset{\boldsymbol{x} \in \mathbb{R}^m}{\text{minimize}} \quad \sum_{i=1}^p f_i(\boldsymbol{x}) + g(\boldsymbol{x}). \tag{2.10}$$

The generalized forward-backward splitting (GFBS) scheme in Algorithm 4 was proposed in [Raguet et al., 2013] for solving (2.10) using the proximity operator of each f_i. This algorithm will be used as a comparison in Section 3.6 with $p = 2$ and $\operatorname{dom}(f_1) = \mathbb{R}^m$. The convergence of GFBS in this case is ensured by [Raguet et al., 2013, Theorem 2.1].

Algorithm 4: Generalized forward-backward [Raguet et al., 2013]

Let $\boldsymbol{\omega} \in]0,1[^p$ be such that $\sum_{i=1}^p \omega_i = 1$.
For every $k \in \mathbb{N}$, let $\gamma_k \in \mathbb{R}_+^*$ and $\lambda_k \in \mathbb{R}_+^*$.
Initialization: For every $i \in \{1, \ldots, p\}$, let $\boldsymbol{z}_0^{(i)} \in \mathbb{R}^m$.
Set $\boldsymbol{x}_0 = \sum_{i=1}^p \omega_i \boldsymbol{z}_0^{(i)}$.
for $k = 0, 1, \ldots$ **do**
  for $i \in \{1, \ldots, p\}$ **do**
    $\boldsymbol{z}_{k+1}^{(i)} = \boldsymbol{z}_k^{(i)} + \lambda_k \left(\operatorname{prox}_{\frac{\gamma_k}{\omega_i} f_i} \left(2\boldsymbol{x}_k - \boldsymbol{z}_k^{(i)} - \gamma_k \nabla g(\boldsymbol{x}_k) \right) - \boldsymbol{x}_k \right)$;
  end
  $\boldsymbol{x}_{k+1} = \sum_{i=1}^p \omega_i \boldsymbol{z}_k^{(i)}$;
end

2.2.2.3 Dual forward-backward algorithm

If f is the zero function and $g = \frac{1}{2}\|\cdot - \boldsymbol{y}\|^2$ for some $\boldsymbol{y} \in \mathbb{R}^m$, then, (2.8) can be re-written as follows,

$$\underset{\boldsymbol{x} \in \mathbb{R}^m}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 + h(\boldsymbol{L}\boldsymbol{x}), \tag{2.11}$$

In the dual problem associated to (2.11), h^* and $\boldsymbol{L}$ are decoupled. Hence, using duality can lead to iterative solvers that only require the proximity operator of h^*, which can easily be derived from the one of h using Moreau's decomposition from Theorem 2.2.9. We present in Algorithm 5 a simplified version of the dual forward-backward (DFB) algorithm proposed in [Combettes et al., 2011] applied to problem (2.11), where we assume that there is no error in the computation of the proximity operator of h^*. The convergence result for Algorithm 5 is given in [Combettes et al., 2011, Theorem 2.2].

Algorithm 5: Dual forward–backward [Combettes et al., 2011] (exact version) applied to problem (2.11)

Set $\rho = |||\boldsymbol{L}|||^{-2}$ and let $\epsilon \in]0, \min\{1, \rho\}[$.
For every $k \in \mathbb{N}$, let $\gamma_k \in [\epsilon, 2\rho - \epsilon]$ and $\lambda_k \in [\epsilon, 1]$.
Initialization: Let $\boldsymbol{u}_0 \in \mathbb{R}^n$.
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{x}_k = \boldsymbol{y} - \boldsymbol{L}^\top \boldsymbol{u}_k$;
  $\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \lambda_k \left(\mathrm{prox}_{\gamma_k h^*}\left(\boldsymbol{u}_k + \gamma_k \boldsymbol{L}\boldsymbol{x}_k\right) - \boldsymbol{u}_k\right)$;
end

It is worth noting that problem (5) is equivalent to computing $\mathrm{prox}_{h(\boldsymbol{L}\cdot)}(\boldsymbol{y})$. Following this remark, in Section 3.6 and Chapter 5 we will use Algorithm 5 in combination with Remark 2.2.8 for computing the proximity operator of a function in a variable metric.

2.2.3 Primal-dual algorithms

In this section we focus on optimization algorithms that make use of both primal and dual updates.

2.2.3.1 Primal-dual splitting algorithm

In [Condat, 2013] and [Vũ, 2013], the authors propose a primal-dual splitting (PDS) algorithm, which relies on a forward-backward scheme to split the differentiable term g and the nonsmooth function f, and which uses duality to decouple h and $\boldsymbol{L}$. Algorithm 6 is a simplified version of the PDS algorithm where we assume that there is no error in the computation of the proximity operators and of the gradient. Note that, if g is the zero function, then we retrieve the algorithm proposed in [Chambolle and Pock, 2011].

The convergence result for the primal sequence in Algorithm 6 is given in [Condat, 2013, Theorem 3.1]. We use Algorithm 6 as a comparison method in Section 3.6.

Algorithm 6: Primal-dual splitting [Condat, 2013; Vũ, 2013] (exact version)

Let $\sigma \in \mathbb{R}_+^*$, $\tau \in \mathbb{R}_+^*$, and $(\forall k \in \mathbb{N})$ let $\lambda_k \in]0, 2[$.
Initialization: Let $(\boldsymbol{x}_0, \boldsymbol{v}_0) \in \mathbb{R}^m \times \mathbb{R}^n$.
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{y}_k = \operatorname{prox}_{\tau f}\left(\boldsymbol{x}_k - \tau\left(\nabla g(\boldsymbol{x}_k) + \boldsymbol{L}^\top \boldsymbol{v}_k\right)\right)$;
  $\boldsymbol{u}_k = \operatorname{prox}_{\sigma h^*}\left(\boldsymbol{v}_k + \sigma \boldsymbol{L}(2\boldsymbol{y}_k - \boldsymbol{x}_k)\right)$;
  $\boldsymbol{x}_{k+1} = \lambda_k \boldsymbol{y}_k + (1 - \lambda_k)\boldsymbol{x}_k$;
  $\boldsymbol{v}_{k+1} = \lambda_k \boldsymbol{u}_k + (1 - \lambda_k)\boldsymbol{v}_k$;
end

2.2.3.2 Accelerated primal-dual splitting algoritm

In [Chambolle and Pock, 2011], the authors proposed an accelerated version of the previous algorithm when g is the zero function and f is *strongly convex*, i.e. there exists $\beta_f \in \mathbb{R}_+^*$ such that for every $\boldsymbol{x} \in \operatorname{dom}(\partial f)$,

$$(\forall \boldsymbol{u} \in \partial f(\boldsymbol{x}))(\forall \boldsymbol{y} \in \mathbb{R}^m) \quad f(\boldsymbol{x}) + \langle \boldsymbol{u}, \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\beta_f}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 \leq f(\boldsymbol{y}).$$

Then, problem (2.8) reads

$$\underset{\boldsymbol{x} \in \mathbb{R}^m}{\text{minimize}} \quad f(\boldsymbol{x}) + h(\boldsymbol{L}\boldsymbol{x}). \tag{2.12}$$

The accelerated primal-dual splitting (A-PDS) algorithm is given in Algorithm 7. The convergence of this algorithm is proven in [Chambolle and Pock, 2011, Theorem 2].

Algorithm 7: Accelerated primal-dual splitting [Chambolle and Pock, 2011, Algorithm 2]

Initialization: Let $\tau_0 \in \mathbb{R}_+^*$ and $\sigma_0 = (\tau_0 |||\boldsymbol{L}|||^2)^{-1}$.
Set $(\boldsymbol{x}_0, \boldsymbol{y}_0) \in \mathbb{R}^m \times \mathbb{R}^n$ and $\overline{\boldsymbol{x}}_0 = \boldsymbol{x}_0$.
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{y}_{k+1} = \operatorname{prox}_{\sigma_k h^*}\left(\boldsymbol{y}_k + \sigma_k \boldsymbol{L}\overline{\boldsymbol{x}}_k\right)$;
  $\boldsymbol{x}_{k+1} = \operatorname{prox}_{\tau_k f}\left(\boldsymbol{x}_k - \tau_k \boldsymbol{L}^\top \boldsymbol{y}_{k+1}\right)$;
  $\theta_k = (1 + 2\beta_f \tau_k)^{-\frac{1}{2}}$, $\tau_{k+1} = \theta_k \tau_k$, $\sigma_{k+1} = \sigma_k / \theta_k$;
  $\overline{\boldsymbol{x}}_{k+1} = \boldsymbol{x}_{k+1} + \theta_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)$;
end

Since the function $f = \frac{1}{2}\|\cdot\|^2$ is strongly convex, in Section 3.7 we combine Algorithm 7 with Remark 2.2.8 to compute the proximity operator of a function in a variable metric.

2.2.3.3 Alternating direction method of multipliers

Finally, we consider the following problem,

$$\underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}} \;\; f(\boldsymbol{J}\boldsymbol{x}) + g(\boldsymbol{K}\boldsymbol{x}) + h(\boldsymbol{L}\boldsymbol{x}), \tag{2.13}$$

with $\boldsymbol{J} \in \mathbb{R}^{p\times m}$, $\boldsymbol{K} \in \mathbb{R}^{q\times m}$ and $\boldsymbol{L} \in \mathbb{R}^{r\times m}$. If $\boldsymbol{J} = \boldsymbol{K} = \boldsymbol{I}_m$, then we recover our starting problem (2.8). Problem (2.13) can be reformulated as follows,

$$\begin{array}{ll} \underset{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{R}^m\times\mathbb{R}^n}{\text{minimize}} & \tilde{f}(\boldsymbol{y}) \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{0}_n \end{array} \tag{2.14}$$

where $n = p + q + r$,

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{J} \\ \boldsymbol{K} \\ \boldsymbol{L} \end{pmatrix}, \quad \boldsymbol{B} = -\boldsymbol{I}_n,$$

and where every $\boldsymbol{y} \in \mathbb{R}^n$ can be decomposed into a triplet $(\boldsymbol{y}^{(p)}, \boldsymbol{y}^{(q)}, \boldsymbol{y}^{(r)}) \in \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^r$, such that

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}^{(p)} \\ \boldsymbol{y}^{(q)} \\ \boldsymbol{y}^{(r)} \end{pmatrix}, \quad \text{and} \quad \tilde{f}(\boldsymbol{y}) = f\left(\boldsymbol{y}^{(p)}\right) + g\left(\boldsymbol{y}^{(q)}\right) + h\left(\boldsymbol{y}^{(r)}\right).$$

Problem (2.14) can be solved with the alternating direction method of multipliers (ADMM) [Gabay and Mercier, 1976; Fortin and Glowinski, 2000], which can be viewed as a primal-dual algorithm [Komodakis and Pesquet, 2015]. The convergence of Algorithm 8 is obtained from [Eckstein and Bertsekas, 1992, Theorem 8].

Algorithm 8: Alternating direction method of multipliers for (2.14)

Let $\boldsymbol{A}$ have full column rank and take $\mu \in \mathbb{R}_+^*$.
Initialization: Let $\boldsymbol{\lambda}_0 \in \mathbb{R}^n$ and $\boldsymbol{y}_0 \in \mathbb{R}^n$.
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{x}_{k+1} = \left(\boldsymbol{A}^\top \boldsymbol{A}\right)^{-1} \boldsymbol{A}^\top (\boldsymbol{\lambda}_k - \boldsymbol{B}\boldsymbol{y}_k)$;
  $\boldsymbol{y}_{k+1} = \operatorname{prox}_{\tilde{f}/\mu}(\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{\lambda}_k)$;
  $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{B}\boldsymbol{y}_{k+1}$;
end

Although the proximity operator of $\tilde{f}/\mu$ is separable with respect to $\boldsymbol{y}^{(p)}$, $\boldsymbol{y}^{(q)}$, and $\boldsymbol{y}^{(r)}$, the computation of the inverse of $\boldsymbol{A}^\top \boldsymbol{A}$ is a serious shortcoming of this algorithm. Nonetheless, reformulation (2.14) is not unique. For

instance, another formulation is presented in [Iordache et al., 2012], where the inverse of $\boldsymbol{A}^\top \boldsymbol{A}$ is easier to compute, but where $\operatorname{prox}_{\tilde{f}/\mu}$ is not separable. In practice, alternating the minimization on the components of $\boldsymbol{y}$ may perform well [Iordache et al., 2012], but the convergence is not guaranteed in this case [Chen et al., 2016]. We use Algorithm 8 in Sections 3.6 and 3.7 as a comparison method.

When addressing (2.8) directly is too difficult, an interesting strategy consists in replacing the original problem by a sequence of intermediate problems that are easier to solve. This idea is at the core of the approaches presented in the next sections, namely the majorize-minimize principle and interior point methods.

2.2.4 Majorize-Minimize Principle

The majorize-minimize principle (MM) was first introduced in [Ortega and Rheinboldt, 1970]. This principle is built on the notion of *majorant*, which is defined hereafter.

Definition 2.2.19 (Majorant) Let $f : \mathbb{R}^m \to]-\infty, +\infty]$. The function $h : \mathbb{R}^m \times \mathbb{R}^m \to]-\infty, +\infty]$ is said to be a majorant of f at $\boldsymbol{y} \in \mathbb{R}^m$ if

$$h(\boldsymbol{y}, \boldsymbol{y}) = f(\boldsymbol{y}) \quad \text{and} \quad (\forall \boldsymbol{x} \in \operatorname{dom}(f)) \;\; h(\boldsymbol{x}, \boldsymbol{y}) \geq f(\boldsymbol{x}).$$

There exist different strategies for designing a majorant. For instance, if the function f is differentiable, then a quadratic majorant of f can be created as follows [Böhning and Lindsay, 1988]

$$(\forall \boldsymbol{x} \in \mathbb{R}^m)(\forall \boldsymbol{y} \in \mathbb{R}^m)$$
$$h(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{y})^\top A(\boldsymbol{y}) (\boldsymbol{x} - \boldsymbol{y}),$$

where $A(\boldsymbol{y}) \in \mathfrak{S}_m^+$ is such that the conditions in Definition 2.2.19 are satisfied. In addition, if f is Lipschitz differentiable with constant $L \in \mathbb{R}_+$, then $A(\cdot)$ can be set equal to $\alpha \boldsymbol{I}_m$ with $\alpha \geq L$. More generally, A can be chosen such that $A(\boldsymbol{y}) - \nabla^2 f(\boldsymbol{y}) \in \mathfrak{S}_m^+$ [Hunter and Lange, 2004]. As shown in [Chouzenoux et al., 2016], this strategy can be used for designing efficient preconditioning matrices in the VMFB algorithm. We will follow this approach in Chapters 3 and 4 for choosing the involved variable metrics.

Finding a minimizer of the majorant can be easier than computing a solution to the original optimization problem. For instance, if the majorant is quadratic, then its minimizer has an explicit expression. Therefore, the MM principle consists in the following approach, instead of minimizing f directly, a sequence of iterates is produced where each iterate is a minimizer

of a majorant of f taken at the previous iterate. This leads to the following scheme, where for every iteration $k \in \mathbb{N}$ the update rule reads

$$\boldsymbol{x}_{k+1} = \underset{\boldsymbol{x} \in \mathrm{dom}\,(f)}{\mathrm{argmin}} \; h(\boldsymbol{x}, \boldsymbol{x}_k),$$

with h a majorant of f at $\boldsymbol{x}_k$. The convergence of this algorithm was established under some assumptions [Jacobson and Fessler, 2007]. Furthermore, this approach has demonstrated good performance in various applications, such as image recovery [Sotthivirat and Fessler, 2002; Erdogan and Fessler, 1999], denoising [Selesnick, 2012] or quantile regression [Hunter and Lange, 2000]. It is worth noting that this algorithm can be applied in a nonconvex setting [Chouzenoux et al., 2013a]. In Chapter 5, we derive an algorithm based on the MM principle to solve a nonconvex minimization problem.

In the next section, we present a specific class of methods used to address constrained optimization problems.

2.2.5 Interior point methods

Interior point methods (IPMs) encompass a wide range of methods which share the ability to handle constraints [Boyd and Vandenberghe, 2004, Chapter 11]. In this section we introduce the fundamental ideas and interior point methods that are used in Chapters 3 and 4.

IPMs became very popular from 1980 onwards due to the polynomial-time projection algorithm proposed in [Karmarkar, 1984], and which was later reinterpreted as an IPM in [Gill et al., 1986]. These approaches then demonstrated excellent performance in linear programming [Wright, 1992]. They were extended to nonlinear problems [Forsgren et al., 2002] thanks to the notion of self-concordance [Nesterov and Nemirovskii, 1994], which will be explained hereafter. IPMs have been applied to a wide variety of applications, including large-scale portfolio optimization [Gondzio and Grothey, 2005], semidefinite programming [Alizadeh, 1995] and image reconstruction [Kim et al., 2007]. Although IPMs are well-established methods, there are still new developments to be made regarding these approaches [Gondzio, 2012; Ge et al., 2019].

2.2.5.1 Principle

Let us consider the following variational formulation,

$$\underset{\boldsymbol{x} \in \mathcal{C}}{\mathrm{minimize}} \;\; f(\boldsymbol{x}), \tag{2.15}$$

where the feasible set $\mathcal{C}$ is defined using p inequality constraints,

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \ldots, p\}) \;\; c_i(\boldsymbol{x}) \leq 0\}. \tag{2.16}$$

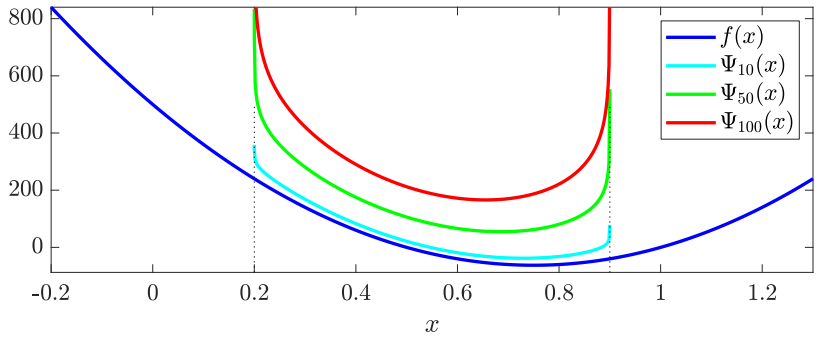


Figure 2.4: *Illustration of the merit function and influence of the barrier parameter when the barrier is chosen as the logarithmic one. In this example, the feasible set is taken as* $\mathcal{C} = \{x \in \mathbb{R} \mid 0.2 \leq x \leq 0.9\}$.

Barrier methods are standard IPMs in which the original problem (2.15) is replaced by a sequence of intermediate unconstrained problems $(\mathcal{P}_{\mu_k})_{k\in\mathbb{N}}$ parametrized by a sequence of *barrier coefficients* $(\mu_k)_{k\in\mathbb{N}}$. For every $\mu \in \mathbb{R}_+^*$, $\mathcal{P}_\mu$ is defined as

$$\mathcal{P}_\mu : \underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}} \;\; f(\boldsymbol{x}) + \mu\mathcal{B}(\boldsymbol{x}), \tag{2.17}$$

where $\mathcal{B}$ is a *barrier function*. The objective function in (2.17) is called the *merit function* and, for every $\mu \in \mathbb{R}_+^*$, it is noted $\Psi_\mu = f + \mu\mathcal{B}$. The barrier can be seen as an approximation of the indicator function of the feasible set; its purpose is to ensure that the constraints involved in $\mathcal{C}$ are always strictly satisfied. As illustrated in Figure 2.4, it should be chosen such that the continuity properties of the constraints are preserved and such that it goes to infinity at the border of the feasible set [Forsgren et al., 2002]. There exist various barrier functions in the literature [Carroll, 1961; Kowalik, 1966], the most popular one being the logarithmic barrier, defined as follows

$$\begin{array}{rcl} \mathcal{B} : \mathbb{R}^m & \to &]-\infty, +\infty] \\ \boldsymbol{x} & \mapsto & \begin{cases} -\displaystyle\sum_{i=1}^{p} \ln(-c_i(\boldsymbol{x})) \text{ if } (\forall i \in \{1,\dots,p\})\ c_i(\boldsymbol{x}) < 0, \\ +\infty \qquad\qquad\qquad\quad \text{otherwise.} \end{cases} \end{array}$$

The logarithmic barrier has interesting features, in particular it is *self-concordant* for every affine or quadratic constraints [Boyd and Vandenberghe, 2004, Chapter 9]. This means that $\mathcal{B}$ is three times continuously differentiable on its domain and that

$$(\forall \boldsymbol{x} \in \operatorname{dom}(\mathcal{B})) \;\; |\mathcal{B}'''(\boldsymbol{x})| \leq 2\mathcal{B}''(\boldsymbol{x})^{3/2}.$$

In addition, if Assumption 2.2.20 below holds and if the barrier is chosen as the logarithmic one, then the sublevel sets of the merit function are compact. This result is detailed in Theorem 2.2.21 hereafter.

Assumption 2.2.20 *Consider problem (2.15), and assume that* $f \in \Gamma_0(\mathbb{R}^m)$ *and* $(\forall i \in \{1, \ldots, p\})$ $c_i \in \Gamma_0(\mathbb{R}^m)$. *Let* $\mathcal{B}$ *be the logarithmic barrier associated with the set* $\mathcal{C}$ *defined in (2.16). Assume in addition that* $\operatorname{dom}(\mathcal{B})$ *is nonempty,* $\mathcal{C} \subset \operatorname{dom}(f)$, *and that the set of solutions to problem (2.15) is nonempty and bounded.*

Theorem 2.2.21 *[Wright, 1992, Theorem 4][2] Under Assumption 2.2.20, for every* $(\mu, \tau) \in \mathbb{R}_+^* \times \mathbb{R}$, *the* τ*-lower level set of* $\Psi_\mu = f + \mu\mathcal{B}$, *defined by*

$$\operatorname{lev}_{\leq \tau}(\Psi_\mu) = \{\boldsymbol{x} \in \operatorname{dom}(\mathcal{B}) \mid \Psi_\mu(\boldsymbol{x}) \leq \tau\},$$

is compact.

From now on, $\mathcal{B}$ will refer to the logarithmic barrier. The path-following method presented hereafter is a specific barrier method where each intermediate problem $\mathcal{P}_{\mu_k}$ is solved before decreasing the barrier coefficient μ_k.

2.2.5.2 Path-following method

If there exists a solution to every intermediate problem (2.17), then one can define a trajectory $\boldsymbol{x}^*$ where, for every $\mu \in \mathbb{R}_+^*$, $\boldsymbol{x}^*(\mu)$ is a solution to $\mathcal{P}_\mu$. The standard *path-following* interior point method detailed in Algorithm 9 follows this trajectory called *central path*. A general convergence result regarding this algorithm is given in Theorem 2.2.22.

Algorithm 9: Path-following barrier method

Let $\rho > 1$ and $\epsilon \in \mathbb{R}_+^*$.
Initialization: Take $\boldsymbol{x}_0 \in \operatorname{dom}(\mathcal{B})$ and $\mu_0 \in \mathbb{R}_+^*$.
for $k = 0, 1, \ldots$ **do**
  Find $\boldsymbol{x}^*(\mu_k)$ a solution to $\mathcal{P}_{\mu_k}$ using an iterative solver initialized with $\boldsymbol{x}_k$;
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}^*(\mu_k)$;
  Stop if $p\mu_k \leq \epsilon$;
  $\mu_{k+1} = \mu_k / \rho$;
end

Theorem 2.2.22 *[Wright, 1992, Theorem 5 (iv)][2] Under Assumption 2.2.20, if* $(\boldsymbol{x}_k)_{k \in \mathbb{N}}$ *is a sequence generated by Algorithm 9, then there exists at least one convergent subsequence, and every of its cluster point is a solution to problem (2.15).*

[2] It is straightforward to extend the original theorem, which was established assuming continuous functions, to l.s.c. functions.

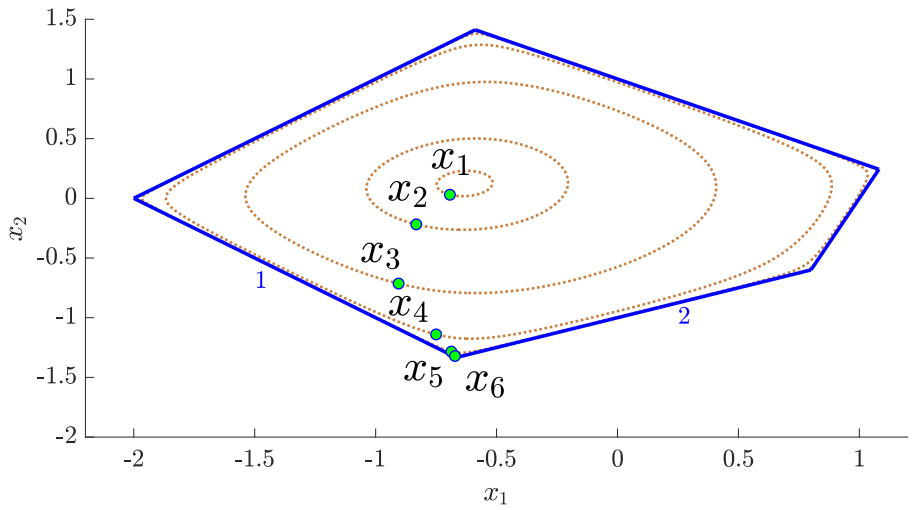


Figure 2.5: *Illustration of Algorithm 9 applied to problem (2.18) with* $\boldsymbol{c} = (1, 2)^\top$*, and where the feasible set is represented as a blue polytope. Hyperparameters are chosen such that* $\mu_0 = 10$*,* $\rho = 4$ *and* $\epsilon = 0.02$*. Brown dotted lines show the iso-*$\mathcal{B}$ *contours.*

We illustrate in Figure 2.5 the central path trajectory given by Algorithm 9 applied to the following linear programming problem,

$$\begin{array}{ll} \underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in]-\infty, 0]^p, \end{array} \tag{2.18}$$

where $\boldsymbol{c} \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{p\times m}$ and $\boldsymbol{b} \in \mathbb{R}^p$. This problem will be used in Section 3.7 and serves as an illustration in the remaining of this section.

The Lagrangian theory provides an interesting reinterpretation of the path-following method.

Link with the Lagrangian theory

We consider problem (2.18) and make the following assumption.

Assumption 2.2.23 *Consider problem (2.18) and assume that there exists* $\boldsymbol{x} \in \mathbb{R}^m$ *such that* $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in]-\infty, 0[^p$*. In addition, assume that for every* $\tau \in \mathbb{R}$*, the sublevel set* $\{\boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in]-\infty, 0]^p \text{ and } \boldsymbol{c}^\top \boldsymbol{x} \leq \tau\}$ *is bounded.*

Under Assumption 2.2.23, Slater's constraint qualification is satisfied, and finding a solution $\boldsymbol{x}^*$ to problem (2.18) is equivalent to finding a saddle point of the Lagrangian $\mathcal{L}$, which is defined as follows,

$$(\forall \boldsymbol{x} \in \mathbb{R}^m)(\forall \boldsymbol{\lambda} \in [0, +\infty[^p) \quad \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

Saddle points of $\mathcal{L}$ are characterized by the KKT conditions; for every solution $\boldsymbol{x}^*$ to problem (2.18), there exists $\boldsymbol{\lambda}^* \in \mathbb{R}^p$ such that the following system of equations is satisfied,

$$\begin{cases} \boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda}^* = \boldsymbol{0}_p, & (2.19) \\ (\forall i \in \{1, \dots, p\})\ \lambda_i^* \geq 0 \text{ and } (\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b})_i \leq 0, & (2.20) \\ (\forall i \in \{1, \dots, p\})\ \lambda_i^* (\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b})_i = 0. & (2.21) \end{cases}$$

For every $\mu \in \mathbb{R}_+^*$, $\mathcal{P}_\mu$ can be rewritten as follows,

$$\begin{array}{ll} \underset{(\boldsymbol{x},\boldsymbol{z}) \in \mathbb{R}^m \times \mathbb{R}^p}{\text{minimize}} & \boldsymbol{c}^\top \boldsymbol{x} + \mu \tilde{\mathcal{B}}(\boldsymbol{z}) \\ \text{s.t.} & \boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}, \end{array}$$

where $\tilde{\mathcal{B}}(\boldsymbol{z}) = -\sum_{i=1}^p \ln(-z_i)$ if $\boldsymbol{z} \in]-\infty, 0[^p$, $+\infty$ otherwise. Finding the solution $\boldsymbol{x}^*(\mu)$ to $\mathcal{P}_\mu$ is equivalent to identifying a saddle point of the associated Lagrangian $\mathcal{L}_\mu$ defined by

$$(\forall \boldsymbol{x} \in \mathbb{R}^m)(\forall \boldsymbol{z} \in \mathbb{R}^p)(\forall \boldsymbol{\lambda} \in \mathbb{R}^p)$$
$$\mathcal{L}_\mu(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \mu \tilde{\mathcal{B}}(\boldsymbol{z}) + \boldsymbol{\lambda}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} - \boldsymbol{z}).$$

There exist $\boldsymbol{z}^* \in \mathbb{R}^p$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^p$ such that the following KKT conditions are satisfied

$$\begin{cases} \boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda}^* = \boldsymbol{0}_p, & (2.22) \\ (\forall i \in \{1, \dots, p\})\ (\boldsymbol{A}\boldsymbol{x}^*(\mu) - \boldsymbol{b})_i < 0, & (2.23) \\ (\forall i \in \{1, \dots, p\})\ \lambda_i^* (\boldsymbol{A}\boldsymbol{x}^*(\mu) - \boldsymbol{b})_i = -\mu, & (2.24) \\ \boldsymbol{z}^* = \boldsymbol{A}\boldsymbol{x}^*(\mu) - \boldsymbol{b}. & (2.25) \end{cases}$$

In the above system of equations, the complementary slackness from (2.21) is replaced by condition (2.24). Hence, the path-following method consists in finding solutions to modified KKT conditions such that the iterates produced by Algorithm 9 almost satisfy (2.19)-(2.21) when μ is small.

It is worth noting that *strict complementarity* always holds in linear programming [Bonnans and Shapiro, 2013, Theorem 3.133], i.e. there exists $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ a solution to the KKT conditions (2.19)-(2.21) such that, for every $i \in \{1, \dots, p\}$, if $(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b})_i = 0$ then $\lambda_i^* > 0$.

Remark 2.2.24 (Strict complementarity) In a more general context than that of linear programming, strict complementarity is not always necessary, see for instance [Jittorntrum, 1984], [Bonnans et al., 2006, Chapter 18] and the concept of *linear monotone complementarity* used in [Bonnans et al., 2006, Chapter 20] for quadratic programming. However, this assumption is often made for deriving convergence results regarding IPMs [Wright, 1992]. Note that this assumption will also be used in Chapter 3.

Algorithm 9 assumes that one can easily compute a solution to each intermediate problem. We present hereafter the Newton barrier method, which uses Newton's method at each iteration to find a solution to $\mathcal{P}_{\mu_k}$. The Newton barrier method is used in particular in Section 3.7.

2.2.5.3 Newton barrier method

Algorithm 10: Newton barrier method

Let $\rho > 1$, $\rho_n > 1$, $\epsilon \in \mathbb{R}_+^*$, $\epsilon_n \in \mathbb{R}_+^*$ and $\alpha \in]0, \frac{1}{2}[$.
Initialization: Take $\boldsymbol{x}_0 \in \mathrm{dom}\,(\mathcal{B})$ and $\mu_0 \in \mathbb{R}_+^*$.
Set $\mu \leftarrow \mu_0$.
for $k = 0, 1, \ldots$ **do**
  $\Delta \boldsymbol{x}_k = -\nabla^2 \Psi_\mu(\boldsymbol{x}_k)^{-1} \nabla \Psi_\mu(\boldsymbol{x}_k)$ (Newton's direction);
  $d_k = -\nabla \Psi_\mu(\boldsymbol{x}_k)^\top \Delta \boldsymbol{x}_k$ (Newton's decrement);
  if $d_k/2 \leq \epsilon_n$ **then**
    Stop if $p\mu \leq \epsilon$;
    $\mu \leftarrow \mu/\rho$;
  else
    $\gamma \leftarrow 1$;
    while $\Psi_\mu(\boldsymbol{x}_k + \gamma \Delta \boldsymbol{x}_k) > \Psi_\mu(\boldsymbol{x}_k) - \alpha\gamma d_k$ **do**
      $\gamma \leftarrow \gamma/\rho_n$ (Backtracking);
    end
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \gamma \Delta \boldsymbol{x}_k$;
  end
end

The Newton barrier method is presented in Algorithm 10. In problem (2.18), the cost is linear and the constraints are affine, hence the merit function is self-concordant. This important property leads to the following result regarding the convergence rate of Algorithm 10.

Theorem 2.2.25 *[Boyd and Vandenberghe, 2004, Section 11.5.3] Consider problem (2.18) and assume that Assumption 2.2.23 is satisfied. Let $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ be generated by Algorithm 10. Then,*

$$(\forall k \geq k_\epsilon) \quad \|\boldsymbol{c}^\top \boldsymbol{x}_k - \boldsymbol{c}^\top \boldsymbol{x}^*\| \leq \epsilon,$$

where k_ϵ is defined by

$$k_\epsilon = \left\lceil \frac{\ln(p\mu_0/\epsilon)}{\ln(\rho)} \right\rceil \left(\frac{p(\rho - 1 - \ln(\rho))}{\eta_1} + \eta_2 \right),$$

with

$$\eta_1 = \frac{\alpha(1-2\alpha)^2}{\rho_n(20-8\alpha)} \quad \textit{and} \quad \eta_2 = \log_2(\log_2(1/\epsilon_n)).$$

It is worth noting that the Newton direction can also be derived from the modified KKT conditions by replacing $\boldsymbol{\lambda}^*$ in (2.22) by its expression given in (2.24), and then applying Newton's method on the resulting equation. Following this remark, it is also possible to apply Newton's method directly on the modified KKT conditions without first eliminating $\boldsymbol{\lambda}^*$. This approach leads to *primal-dual* barrier methods with search directions on both the primal and dual variables. In particular, one can mention the infeasible primal-dual path-following algorithm presented in [Gondzio, 2012] which has polynomial complexity.

We introduce one last concept related to IPMs, which is the *analytic center*.

2.2.5.4 Analytic center

Definition 2.2.26 (Analytic center) Let $\mathcal{C}$ by defined as in (2.16) and let $\mathcal{K}$ be defined by a set of equalities,

$$\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \dots, q\}) \;\; h_i(\boldsymbol{x}) = 0\},$$

where $(\forall i \in \{1, \dots, q\})\ h_i : \mathbb{R}^m \to]-\infty, +\infty]$. Let $\mathcal{B}$ be the logarithmic barrier associated to $\mathcal{C}$. If there exists a unique solution to the following problem,

$$\underset{\boldsymbol{x} \in \mathcal{K}}{\text{minimize}} \;\; \mathcal{B}(\boldsymbol{x}),$$

then this point is called the *analytic center* of $\mathcal{C} \cap \mathcal{K}$.

This notion has led to a specific class of IPMs called *analytic center methods* [Sonnevend, 1986] [Jarre, 1992, Section 2.1]. The analytic center of a set of inequalities can be thought of as the feasible point which maximizes the product of its margins with regards to the boundaries of this set [Boyd and Vandenberghe, 2004, Section 8.5.3]. It is worth noting that the analytic center depends on how the feasible set is specified. Adding a redundant inequality in the definition of $\mathcal{C}$ will change the position of the analytic center. Figure 2.6 illustrates this observation.

Consider problem (2.15) and let $\mathcal{J}_P$ be the set of indices $i \in \{1, \dots, p\}$ such that there exists a solution to (2.15) for which the ith constraint is not active. In addition, let $\tilde{\mathcal{C}}$ and $\mathcal{K}$ be defined as follows,

$$\tilde{\mathcal{C}} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \mathcal{J}_P)\ c_i(\boldsymbol{x}) < 0\},$$

$$\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \dots, p\} \setminus \mathcal{J}_P)\ c_i(\boldsymbol{x}) = 0 \;\text{ and }\; f(\boldsymbol{x}) = f^*\},$$

where f^* is the minimum value of f in $\mathcal{C}$. Then, if it exists, the analytic center of the set $\tilde{\mathcal{C}} \cap \mathcal{K}$ is a solution to problem (2.15). When it exists, this specific solution is referred to as the analytic center of the solution set to (2.15). This insightful characterization will be used in Chapter 3.

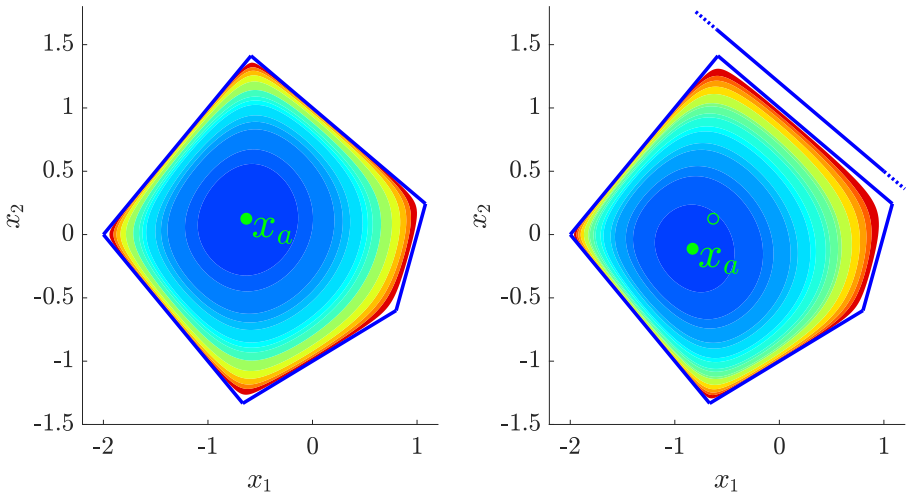


Figure 2.6: *Left: analytic center $\boldsymbol{x}_a$ of the feasible set defined by the affine blue constraints. Right: influence of a redundant constraints in the definition of the feasible set on the position of the analytic center. Colors represent the logarithmic barrier value.*

In this section, we have presented classical IPMs which have proven their effectiveness, in particular for linear and quadratic programming. However, when dealing with more general optimization problems, these methods have some notable drawbacks. For instance, they require the inversion of a linear system at each iteration, and they are usually restricted to twice-differentiable objective functions. In addition, results about their pointwise convergence might be limited, as illustrated in Theorem 2.2.22. We address these issues in Chapter 3, where we propose to combine the barrier method with proximal tools.

§ 2.3 Summary

In this chapter, we have introduced a mathematical formulation for inverse problems, as well as three resolution methods investigated in the next chapters. The notation and mathematical definitions used throughout this document have been presented in Section 2.2.1. As mentioned previously, the goal of this thesis is to illustrate the versatility of the proximity operator in the context of inverse problems, by combining it with each one of the three aforementioned approaches.

In Chapter 3, we propose a variational method based on a new algorithm combining the proximity operator with the barrier method, which has been presented in Section 2.2.5. Chapter 4 is dedicated to the study of an

optimization-inspired neural network. The proposed architecture is obtained by applying the concept of deep unfolding presented in Section 2.1.4.3, on a proximal interior point iterative scheme. Lastly, in Chapter 5 we consider a Bayesian setting and propose to accelerate a hybrid Gibbs sampler with a new sampling algorithm based on the proximity operator.

- Chapter 3 -

A fast proximal interior point algorithm for constrained variational formulations

As pointed out in Section 2.1.2, many problems in image processing, such as segmentation [Chan and Vese, 2001], classification [Briceño-Arias et al., 2017], or restoration [Malgouyres, 2002; Nikolova, 2004], can be formulated as the minimization of a convex objective function under convex constraints. Such problem can be successfully addressed by interior point methods, which have been presented in Section 2.2.5. From a numerical perspective, IPMs have demonstrated very good performance on several challenging applications, such as image reconstruction and multispectral image unmixing [Johnson et al., 2000; Chouzenoux et al., 2014a]. However, it is worth noting that most of interior point approaches rely on first or second-order methods and, therefore, assume that the objective function is at least twice-differentiable [Armand et al., 2000; Bonettini and Serafini, 2009a]. This characteristic restricts their scope of application since the quality of the solution to an inverse problem and its robustness to noise, can often be improved by including a nondifferentiable regularization term in the objective function. Although IPMs can handle the ℓ_1 norm [Fu et al., 2006; Kim et al., 2007; Fountoulakis and Gondzio, 2016], for more general non-smooth penalizations, optimization approaches relying on the proximity operator seem more appropriate [Combettes and Pesquet, 2011]. In this chapter, we propose to combine a proximal algorithm with the logarithmic barrier method, leading to a proximal interior point algorithm referred to as PIPA. One advantage of PIPA is that it can handle nonsmooth functions while ensuring that the constraints are always satisfied. In addition, PIPA includes a variable metric that can be used to boost its speed of convergence.

This chapter is organized as follows. After describing the challenges involved and motivating our approach in Section 3.1, we introduce the variational formulation and our assumptions in Section 3.2. The proposed method is then presented in Section 3.3 and summarized in Algorithms 11 and 12.

Our main theoretical results are provided in Section 3.4, while the proofs for these results are given in Section 3.5. Numerical experiments are presented in Sections 3.6 and 3.7.

§ 3.1 Challenges and motivation

Combining a proximal algorithm with the classical barrier method requires to address two main challenges that are discussed in this section: *(i)* handling the barrier term while maintaining acceptable computational complexity and speed, *(ii)* ensuring the convergence of the algorithm without the gradient-Lipschitz property. The first difficulty will be addressed thanks to a splitting strategy and a variable metric, while the second point will be tackled by including an appropriate line search in the proposed algorithm.

In most applications, the objective function is composite, in that it can be split into a nonsmooth term and a differentiable term. In particular, intermediate problems (2.17) arising in the barrier method may have this composite nature. As detailed in Section 2.2.2, some proximal resolution methods take advantage of this decomposition, like the VMFB algorithm, summarized in Algorithm 3, where a variable metric can be used for acceleration or simplification. The KL inequality presented in Section 2.2.1.2 proves to be a central argument for proving the convergence of this algorithm [Chouzenoux et al., 2014b; Frankel et al., 2015], and for deriving convergence rates [Attouch and Bolte, 2009; Attouch et al., 2013].

Proximal and interior point methods can be combined to produce efficient solvers, as illustrated in [Kaplan and Tichatschke, 1998], where the authors proposed an algorithm that minimizes a convex differentiable function over convex inequality constraints. However, this framework does not make use of any splitting strategy, hence it assumes that the proximity operator of the merit function is easy to compute. More recently, a preconditioning strategy based on the logarithmic barrier was proposed in [Valkonen, 2017] to modify the dual update in a proximal primal-dual algorithm.

A major challenge, when dealing with IPMs is that the logarithmic barrier does not satisfy the gradient-Lipschitz property. Therefore, specific line search strategies have to be designed in order to preserve the convergence properties of the methods used to solve the intermediate problems [Chouzenoux et al., 2012]. This question has also been addressed in the context of VMFB algorithms, when the gradient of the smooth term in the objective function is not globally Lipschitz-continuous. Following the work of [Tseng and Yun, 2009], Armijo-type line searches were proposed in [Bello Cruz and Nghia, 2016; Bonettini et al., 2016], where the convergence of the algorithm is obtained in a convex setting under suitable assumptions on the variable metrics. However, the line search strategy in [Bello Cruz and Nghia, 2016] requires multiple gradient computations, while

[Bonettini et al., 2016] requires the domain of the nondifferentiable function to be closed. Other line searches have also been studied in [Salzo, 2017], where convergence guarantees and convergence rate in terms of function values are provided for the convex case. It is worth noting that, similarly to [Bonettini and Prato, 2015b], [Salzo, 2017] considers a relaxed version of the monotonicity condition on the variable metrics from [Combettes and Vũ, 2014], where the metrics converge to a multiple of the identity operator, with a multiplicative factor which is allowed to vary along iterations. This assumption still remains restrictive compared to the conditions required in [Chouzenoux et al., 2014b], where the proof of convergence is carried out using the KL property.

In this chapter, we propose to combine the VMFB algorithm with the logarithmic barrier method, leading to our proposed algorithm PIPA. We will make use of one of the line searches investigated in [Salzo, 2017] to determine the stepsize value for each iteration of the proposed algorithm. Although our assumptions on the function domains are different from those in [Salzo, 2017], we prove that the line search remains valid in our context. Furthermore, we carry out the convergence analysis of the proposed algorithm under a mild boundedness condition on the involved variable metrics. Under some additional assumptions, we derive a linear convergence rate for the inner loop involved in PIPA.

§ 3.2 Optimization problem and assumptions

In this chapter we consider the following constrained minimization problem,

$$\mathcal{P}_0 : \quad \underset{\boldsymbol{x}\in\mathcal{C}}{\text{minimize}} \;\; f(\boldsymbol{x}) + g(\boldsymbol{x}) \tag{3.1}$$

where the feasible set $\mathcal{C}$ is defined as follows,

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \dots, p\}) \;\; c_i(\boldsymbol{x}) \leq 0\}.$$

We also define the set $\mathcal{D}$ as

$$\mathcal{D} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \dots, p\}) \;\; c_i(\boldsymbol{x}) < 0\},$$

and assume that Assumption 3.2.1 below is satisfied.

Assumption 3.2.1

(i) The set of solutions to $\mathcal{P}_0$ is nonempty and bounded.

(ii) Functions f, g and $(c_i)_{1\leq i\leq p}$ belong to $\Gamma_0(\mathbb{R}^m)$, and $f + g$ is bounded from below. The set $\mathcal{D}$ is assumed to be nonempty, open, and $\mathcal{C} \subset \operatorname{dom}(f) \subset \operatorname{dom}(g)$.

(iii) Functions g and $(c_i)_{1\leq i\leq p}$ are differentiable on $\mathcal{C}$; ∇g and $(\nabla c_i)_{1\leq i\leq p}$ are Lipschitz–continuous on any compact subset of $\mathcal{C}$.

Since the functions $(c_i)_{1\leq i\leq p}$ are l.s.c., the closure of $\mathcal{D}$ is equal to $\mathcal{C}$.

Remark 3.2.2 (Role of the constraints) In image processing, the constraints can be derived from the underlying geometry of the problem [Harizanov et al., 2013]. For instance, inequality constraints are used in a problem of deformable image matching in [Musse et al., 2001] to ensure that the estimated image deformation is injective and preserves the topology. Constraints can also serve to enforce some a priori knowledge about the solution, and act as regularization terms, as in the image segmentation approach in [Klodt and Cremers, 2011], where bound constraints are imposed on the segmented areas and their barycenters.

Following the framework of IPMs, we propose to reformulate the constrained problem $\mathcal{P}_0$ as a sequence $(\mathcal{P}_{\mu_j})_{j\in\mathbb{N}}$ of modified subproblems parametrized for every $j \in \mathbb{N}$ by a barrier coefficient $\mu_j > 0$ and defined as

$$\mathcal{P}_{\mu_j}: \quad \underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}} \;\; f(\boldsymbol{x}) + g(\boldsymbol{x}) + \mu_j \mathcal{B}(\boldsymbol{x}) \tag{3.2}$$

where $\mathcal{B}$ is the logarithmic barrier associated with the constraints:

$$\begin{aligned} \mathcal{B}: \mathbb{R}^m &\to \;]-\infty, +\infty] \\ \boldsymbol{x} &\mapsto \begin{cases} -\sum_{i=1}^{p} \ln(-c_i(\boldsymbol{x})) & \text{if } \boldsymbol{x} \in \mathcal{D} \\ +\infty & \text{otherwise.} \end{cases} \end{aligned}$$

For simplicity, for every $\boldsymbol{x} \in \mathbb{R}^m$ we introduce the shorter notation $c(\boldsymbol{x}) = (c_i(\boldsymbol{x}))_{1\leq i\leq p} \in \mathbb{R}^p$, and the following functions,

$$(\forall \mu > 0) \quad \varphi_\mu = g + \mu\mathcal{B}, \quad \text{and} \quad \Psi_\mu = f + \varphi_\mu,$$

where Ψ_μ is designated as the merit function.

§ 3.3 PIPA ALGORITHM

The proposed method, PIPA, is made of two interlocked loops. These are detailed in Algorithms 11 and 12, where $\preceq$ denotes the Loewner partial order for matrices.

Given $j \in \mathbb{N}$, Algorithm 11 produces an approximate solution to $\mathcal{P}_{\mu_j}$ via VMFB iterations consisting in a gradient step on the smooth term φ_{μ_j}, and a proximal step on the nondifferentiable term f. The proximity operators are computed within the metric induced by symmetric definite positive preconditioning matrices, whose eigenvalues are bounded from below

Algorithm 11: $\mathcal{A}_\mu(\boldsymbol{x}_0, \delta, \theta, \overline{\gamma}, \epsilon, \underline{\nu}, \overline{\nu})$

Inputs are such that $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu) \in (\mathbb{R}_+^*)^2$, $\epsilon \in \mathbb{R}_+$ and $0 < \underline{\nu} \leq \overline{\nu}$;
Initialization: $\boldsymbol{x}_0 \in \mathcal{D}$;
for $k = 0, 1, \ldots$ **do**
  Choose $\boldsymbol{M}_k \in \mathfrak{S}_m^+$ such that $\underline{\nu}\boldsymbol{I}_m \preceq \boldsymbol{M}_k \preceq \overline{\nu}\boldsymbol{I}_m$;
  for $l = 0, 1, \ldots$ **do**
    $\tilde{\boldsymbol{x}}_{k,l} = \operatorname{prox}_{\overline{\gamma}\theta^l f}^{\boldsymbol{M}_k}\left(\boldsymbol{x}_k - \overline{\gamma}\theta^l \boldsymbol{M}_k^{-1} \nabla\varphi_{\mu_j}(\boldsymbol{x}_k)\right)$;
    Exit loop if (3.3) is satisfied;
  end
  $\boldsymbol{x}_{k+1} = \tilde{\boldsymbol{x}}_{k,l}$;
  $\gamma_k = \overline{\gamma}\theta^l$;
  $\boldsymbol{v}_{k+1} = \frac{1}{\gamma_k}\boldsymbol{M}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) - \nabla\varphi_\mu(\boldsymbol{x}_k) + \nabla\varphi_\mu(\boldsymbol{x}_{k+1})$;
  Stop if $\|\boldsymbol{v}_{k+1}\| < \epsilon$;
end
Return $(\boldsymbol{x}_{k+1}, \boldsymbol{v}_{k+1})$;

Algorithm 12: Proximal Interior Point Algorithm (PIPA)

Let $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu_0) \in (\mathbb{R}_+^*)^2$, $\boldsymbol{x}_0 \in \mathcal{D}$, and let $(\mu_j)_{j\in\mathbb{N}}$ and $(\epsilon_j)_{j\in\mathbb{N}}$ satisfy Assumption 3.3.1;
for $j = 0, 1, \ldots$ **do**
  Let $0 < \underline{\nu}_j \leq \overline{\nu}_j$;
  $(\boldsymbol{x}_{j+1}, \boldsymbol{v}_{j+1}) = \mathcal{A}_{\mu_j}(\boldsymbol{x}_j, \delta, \theta, \overline{\gamma}, \epsilon_j, \underline{\nu}_j, \overline{\nu}_j)$;
  $\boldsymbol{\lambda}_{j+1} = \left(-\frac{\mu_j}{c_i(\boldsymbol{x}_{j+1})}\right)_{1\leq i\leq p}$;
end
Return $\boldsymbol{x}_{j+1}$;

and from above (using $\underline{\nu}$ and $\overline{\nu}$ positive constants). For well-chosen matrices, this variable metric strategy can significantly improve the convergence speed. Preconditioning matrices can be determined for instance through a majorize-minimize procedure [Chouzenoux et al., 2016].

It must be emphasized that, since the barrier is logarithmic, the gradient of φ_{μ_j} is not Lipschitz-continuous on $\mathbb{R}^m$. Thus, the VMFB algorithm must be associated with a line search to find an appropriate value for the stepsize so as to guarantee convergence of the scheme. In Algorithm 11, we use the backtracking line search method investigated in [Salzo, 2017, LS1], itself a generalization of [Bello Cruz and Nghia, 2016]. At iteration $k \in \mathbb{N}$, this backtracking procedure stops if

$$\varphi_\mu(\tilde{\boldsymbol{x}}_{k,l}) - \varphi_\mu(\boldsymbol{x}_k) - \langle \tilde{\boldsymbol{x}}_{k,l} - \boldsymbol{x}_k, \nabla\varphi_\mu(\boldsymbol{x}_k)\rangle \leq \frac{\delta}{\bar{\gamma}\theta^l}\|\tilde{\boldsymbol{x}}_{k,l} - \boldsymbol{x}_k\|^2_{\boldsymbol{M}_k}, \tag{3.3}$$

for some $l \in \mathbb{N}$. Such line search ensures both sufficient decrease of the criterion and feasibility of the next iterate. It is interesting to note that if the whole cost function in $\mathcal{P}_0$ is smooth ($f = 0$), then it reduces to the standard Armijo line search along the steepest direction. Applications related to ℓ_p-norms, with $1 < p < 2$, and Bregman distances where considered in [Salzo, 2017], but, to the best of our knowledge, it is the first time that this line search is applied in the context of an interior point approach.

The resolution of the inner subproblem through Algorithm 11 is stopped once a certain accuracy is reached. In practice, we propose to stop the iterations once the norm of one element of the sequence $(\boldsymbol{v}_{k+1})_{k\in\mathbb{N}}$, where $(\forall k \in \mathbb{N})\ \boldsymbol{v}_{k+1} \in \partial\Psi_\mu(\boldsymbol{x}_{k+1})$, is sufficiently small. As stated in Assumption 3.3.1 below, the sequence $(\epsilon_j)_{j\in\mathbb{N}}$ and the barrier parameter $(\mu_j)_{j\in\mathbb{N}}$, have to be chosen properly to secure the convergence of the sequence of iterates produced by Algorithm 12 to a solution to the initial problem $\mathcal{P}_0$.

Assumption 3.3.1 (Hyperparameters) *For every $j \in \mathbb{N}$, $\mu_j > 0$ and $\epsilon_j > 0$. In addition, $\lim_{j\to+\infty} \mu_j = 0$ and $\lim_{j\to+\infty} \epsilon_j/\mu_j = 0$.*

Remark 3.3.2 The sequence $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$, produced by Algorithm 12, can be linked to the Lagrangian parameters associated with the constrained problem $\mathcal{P}_0$. This is used in proof of convergence in Section 3.5.3.

3.3.1 Related works

It can be noted that there exist links between the proposed PIPA algorithm and a different class of methods called diagonal or penalization methods [Attouch et al., 2011b; Garrigos et al., 2018], for which a general study was recently provided for the continuous setting in [Attouch et al., 2018]. In [Attouch et al., 2011a], the authors proposed a similar approach in the discrete setting based on the FB algorithm. However, they assume that the

gradient of the *penalization function*, which is the equivalent of our barrier, is Lipschitz continuous, and they let the *penalization parameter* tend to infinity, whereas our barrier parameter decreases to zero. In addition, their method does not solve a problem of the form (3.1), instead, it performs a hierarchical minimization [Attouch et al., 2011a, Eq. (6)]. This key difference is also highlighted in [Alvarez and Cabot, 2006] and [Cabot, 2005], in the continuous and discrete settings, respectively. In [Cabot, 2005] the author studied the proximal point algorithm, without any forward-backward step, and considered a penalization parameter that, similarly to our barrier parameter, vanishes to zero. The optimization problem that is solved by this algorithm then depends on the rate of reduction chosen for this coefficient. In addition, it is assumed that the penalization function is bounded from below, which is not necessarily satisfied by the logarithmic barrier.

Related works also include Bregman distance approaches and entropy-like proximal algorithms [Iusem et al., 1994; Brito et al., 2012; Quiroz et al., 2015], where the Euclidean norm in the definition of the proximity operator is replaced by a divergence measure. The latter can be chosen such that feasibility is ensured at each iteration. However, the computation of the modified proximity operator in such methods is usually not straightforward.

In the next sections, we provide a convergence study of PIPA.

§ 3.4 Main convergence results

In this section we state our main theoretical results.

3.4.1 Well-definedness of Algorithm 11

First, Theorem 3.4.1 guarantees that the stopping criterion in Algorithm 11 is well-defined.

Theorem 3.4.1 *Under Assumption 3.2.1, for every* $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0, \overline{\nu}]$ *and* $\boldsymbol{x}_0 \in \mathcal{D}$, *for every* $\epsilon \in \mathbb{R}_+^*$ *there exists* $k \in \mathbb{N}$ *such that* $\|\boldsymbol{v}_{k+1}\| < \epsilon$.

Proof. See Section 3.5.1.4.
□

3.4.2 Convergence rate of Algorithm 11

An important feature of Algorithm 11 is the decay rate of the sequence $(\|\boldsymbol{v}_{k+1}\|)_{k\in\mathbb{N}}$. This rate can be made explicit for the particular instance of

linear inequality constraints. More precisely, we focus on the case when the constrained problem takes the form:

$$\begin{array}{ll} \underset{\boldsymbol{x}\in\mathbb{R}^m}{\text{minimize}} & \widetilde{g}(\boldsymbol{H}\boldsymbol{x}) + f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in]-\infty, 0]^p, \end{array} \tag{3.4}$$

where $\widetilde{g}\colon \mathbb{R}^n \to]-\infty, +\infty]$, $\boldsymbol{H} \in \mathbb{R}^{n\times m}$, $\boldsymbol{A} \in \mathbb{R}^{p\times m}$, $\boldsymbol{b} \in \mathbb{R}^p$, and the involved functionals satisfy the following assumption.

Assumption 3.4.2

(i) *f is a polyhedral function, i.e. its epigraph a finite intersection of closed halfspaces.*

(ii) *$\widetilde{g}$ is l.s.c. with an open domain, it is strongly convex on any compact subset of* $\operatorname{dom}(\widetilde{g})$ *and it is twice continuously differentiable on* $\operatorname{dom}(\widetilde{g})$.

(iii) *$f + \widetilde{g} \circ \boldsymbol{H}$ is proper and it is continuous on the domain of its subdifferential.*

Remark 3.4.3 Assumption 3.4.2 holds for instance if $\widetilde{g} = \frac{1}{2}\|\cdot - \boldsymbol{y}\|^2$ with $\boldsymbol{y} \in \mathbb{R}^n$ and if $f = \kappa\|\boldsymbol{W}\cdot\|_1$ where $\kappa \geq 0$ and $\boldsymbol{W} \in \mathbb{R}^{q\times m}$ is a linear transform (e.g., a wavelet analysis operator [Pustelnik et al., 1999; Chaux et al., 2010]). This corresponds to an ℓ_1-regularized least-squares problem, at the core of many applications such as denoising [Rudin et al., 1992], image restoration [Fu et al., 2006], machine learning, or biological data analysis [Hastie et al., 2009, Chapter 18.4].

The following result can be deduced from existing results concerning the use of the KL inequality in optimization.

Theorem 3.4.4 *Let $\mu > 0$ and consider the barrier problem $\mathcal{P}_\mu$ associated to problem (3.4). Under Assumptions 3.2.1 and 3.4.2, for every $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \overline{\nu}) \in (\mathbb{R}_+^*)^2$, $\underline{\nu} \in]0, \overline{\nu}]$, and $\boldsymbol{x}_0 \in \mathcal{D}$, the sequence $(\boldsymbol{v}_{k+1})_{k\in\mathbb{N}}$ generated by Algorithm 11 converges linearly to 0 when $\epsilon = 0$.*

Proof. See Section 3.5.2.2.
□

3.4.3 Convergence of Algorithm 12

We finally present results regarding the convergence of the proposed method PIPA, i.e. Algorithm 12.

Theorem 3.4.5 *Suppose that Assumptions 3.2.1 and 3.3.1 hold for every* $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu_0) \in (\mathbb{R}_+^*)^2$ *and* $\boldsymbol{x}_0 \in \mathcal{D}$. *Then, any sequence* $(\boldsymbol{x}_{j+1}, \boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$ *generated by Algorithm 12 is bounded. In addition, every of its cluster point* $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ *is a primal-dual solution to* $\mathcal{P}_0$, *i.e.* $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ *is a saddle point for the Lagrangian defined in* (3.35).

Proof. See Section 3.5.3.1.
□

A stronger convergence result can be obtained under additional assumptions. In particular, the following condition will turn out to play an important role.

Assumption 3.4.6 *Either the constraints are affine, i.e.* $\boldsymbol{c} : \boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$ *where* $\boldsymbol{A} \in \mathbb{R}^{p\times m}$ *and* $\boldsymbol{b} \in \mathbb{R}^p$, *and* $\boldsymbol{A}$ *has full column rank (i.e.* $\boldsymbol{A}$ *is injective), or there exists* $i \in \{1, \dots, p\}$ *such that* c_i *is strictly convex.*

Let $\mathcal{S}_P$ be the set of solutions to $\mathcal{P}_0$ (primal solutions), and let $\mathcal{S}_D$ be the set of solutions to the Lagrange dual problem associated with $\mathcal{P}_0$ (dual solutions), whose definition is recalled below [Boyd and Vandenberghe, 2004, Section 5.2].

$$\underset{\boldsymbol{\lambda}\in[0,+\infty[^p}{\text{maximize}} \quad \inf_{\boldsymbol{x}\in\mathbb{R}^m} f(\boldsymbol{x}) + g(\boldsymbol{x}) + \boldsymbol{\lambda}^\top c(\boldsymbol{x})$$

In addition, let

$$\mathcal{J}_P = \{i \in \{1, \dots, p\} \,|\, (\exists \boldsymbol{x} \in \mathcal{S}_P)\ c_i(\boldsymbol{x}) < 0\}, \tag{3.5}$$

and

$$\mathcal{J}_D = \{i \in \{1, \dots, p\} \,|\, (\exists \boldsymbol{\lambda} \in \mathcal{S}_D)\ \lambda_i > 0\}. \tag{3.6}$$

Our main convergence result, summarized in Theorem 3.4.7, provides a useful characterization for the limit point of Algorithm 12 using the notion of *analytic center* introduced in Section 2.2.5.4. The definition and the proof of existence of the analytic center are given in Section 3.5.3.2.

Theorem 3.4.7 *Under Assumptions 3.2.1 and 3.3.1, the following statements hold.*

(i) If there exists only one element in $\mathcal{S}_P$ *(resp.* $\mathcal{S}_D$*), then the sequence* $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ *(resp.* $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$*) produced by Algorithm 12 converges to this unique primal (resp. dual) solution to* $\mathcal{P}_0$.

(ii) Suppose that there exist at least two distinct elements in $\mathcal{S}_P$ *(resp.* $\mathcal{S}_D$*), that Assumption 3.4.6 holds, and that* $\mathcal{P}_0$ *has the strict complementarity property, i.e.* $\mathcal{J}_P \cup \mathcal{J}_D = \{1, \dots, p\}$. *Then the sequence* $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ *(resp.* $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$*) produced by Algorithm 12 converges to a primal (resp. dual) solution to* $\mathcal{P}_0$, *which is the analytic center of* $\mathcal{S}_P$ *(resp.* $\mathcal{S}_D$*).*

Proof. See Section 3.5.3.3.
□

The next sections providing the proofs for the aforementioned theorems are organized as follows. First, we show in Section 3.5.1.2 that, under the considered assumptions, the chosen line search is well-defined. Then, we derive Lemma 3.5.6, Corollary 3.5.8 and Lemma 3.5.10 in Section 3.5.1.3, which lead to the proof of Theorem 3.4.1 in Section 3.5.1.4. Section 3.5.2 is dedicated to the convergence analysis and convergence rate of Algorithm 11. In Section 3.5.2.1 we start by deriving Lemma 3.5.12 which, together with Lemmas 3.5.6 and 3.5.10, ensures that the sufficient decrease, relative error and continuity conditions required in [Attouch et al., 2013, Theorem 2.9] are satisfied. This leads to Proposition 3.5.13. We then derive Proposition 3.5.14 which directly leads to the proof of Theorem 3.4.4 in Section 3.5.2.2. Finally, we study the convergence of Algorithm 12 in Section 3.5.3 based on a Lagrangian approach. The proof for Theorem 3.4.5 is given in Section 3.5.3.1 and the proof of Theorem 3.4.7 is provided in 3.5.3.3.

§ 3.5 Proofs

3.5.1 Well-definedness of Algorithm 11

3.5.1.1 Preliminary results

First, we show that there exists a solution to every intermediate problem (3.2).

Corollary 3.5.1 *Under Assumption 3.2.1, for every $\mu \in \mathbb{R}^*_+$, the solution set to $\mathcal{P}_\mu$ is a nonempty convex compact subset of $\mathcal{D}$.*

Proof. Let $\mu \in \mathbb{R}^*_+$. By assumption, there exists $\boldsymbol{x}_0 \in \mathcal{D}$ such that $\boldsymbol{x}_0 \in \operatorname{dom}(\Psi_\mu)$. The set $\operatorname{lev}_{\leq \Psi_\mu(\boldsymbol{x}_0)}(\Psi_\mu) = \{\boldsymbol{x} \in \mathcal{D} \mid \Psi_\mu(\boldsymbol{x}) \leq \Psi_\mu(\boldsymbol{x}_0)\}$ is nonempty since it includes $\boldsymbol{x}_0$, it is convex since $\mathcal{D}$ is convex and $\Psi_\mu \in \Gamma_0(\mathbb{R}^m)$, and it is compact in view of Theorem 2.2.21. Solving $\mathcal{P}_\mu$ amounts to minimizing Ψ_μ over $\operatorname{lev}_{\leq \Psi_\mu(\boldsymbol{x}_0)}(\Psi_\mu)$. Hence, the solution set to $\mathcal{P}_\mu$ is nonempty, convex, closed and bounded, as a subset of $\operatorname{lev}_{\leq \Psi_\mu(\boldsymbol{x}_0)}(\Psi_\mu)$ which is compact.
□

Next, we derive a preliminary result about functions φ_μ and Ψ_μ with $\mu > 0$.

Lemma 3.5.2 *Under Assumption 3.2.1(iii), for every $\mu > 0$, $\nabla\varphi_\mu$ is Lipschitz-continuous on every compact subset of $\mathcal{D}$.*

Proof. Let $\mathcal{K}$ be a compact subset of $\mathcal{D}$. By assumption, for every $i \in \{1, \dots, p\}$, c_i is differentiable on $\mathcal{K}$, so it is continuous on $\mathcal{K}$ and, according to the extreme value theorem, it is bounded on $\mathcal{K}$ and it attains its bounds. Thus, there exist $(\underline{c}, \overline{c}) \in]-\infty, 0[^2$ such that $(\forall i \in \{1, \dots, p\})(\forall \boldsymbol{x} \in \mathcal{K})$ $\underline{c} \leq c_i(\boldsymbol{x}) < \overline{c}$. Hence, for every $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{K}^2$,

$$\begin{aligned} \|\nabla \mathcal{B}(\boldsymbol{x}) - \nabla \mathcal{B}(\boldsymbol{y})\| &\leq \sum_{i=1}^{p} \frac{\|c_i(\boldsymbol{y})\nabla c_i(\boldsymbol{x}) - c_i(\boldsymbol{x})\nabla c_i(\boldsymbol{y})\|}{|c_i(\boldsymbol{x})c_i(\boldsymbol{y})|} \\ &\leq \sum_{i=1}^{p} \frac{|c_i(\boldsymbol{y})|}{\underline{c}^2}\|\nabla c_i(\boldsymbol{x}) - \nabla c_i(\boldsymbol{y})\| \\ &\quad + \frac{|c_i(\boldsymbol{y}) - c_i(\boldsymbol{x})|}{\underline{c}^2}\|\nabla c_i(\boldsymbol{y})\|. \end{aligned} \tag{3.7}$$

In addition, by assumption, for every $i \in \{1, \dots, p\}$, ∇c_i is L_i-Lipschitz continuous on $\mathcal{K}$ for some $L_i > 0$; in particular, it is bounded by some constant $K_i > 0$. Hence, for every $i \in \{1, \dots, p\}$, c_i is K_i-Lipschitz continuous on $\mathcal{K}$ and we deduce from (3.7) that

$$\|\nabla \mathcal{B}(\boldsymbol{x}) - \nabla \mathcal{B}(\boldsymbol{y})\| \leq \left(\sum_{i=1}^{p} \frac{\overline{c}L_i + K_i^2}{\underline{c}^2} \right) \|\boldsymbol{x} - \boldsymbol{y}\|.$$

Therefore, for every $\mu > 0$, $\nabla \varphi_\mu = \nabla g + \mu \nabla \mathcal{B}$ is Lipschitz continuous on $\mathcal{K}$. □

3.5.1.2 Line search

We show in this section that the chosen line search is well-defined given our assumptions. Let $(\overline{\gamma}, \mu) \in (\mathbb{R}_+^*)^2$, $\theta \in]0, 1[$ and $\boldsymbol{M} \in \mathfrak{S}_m^+$. Let h be such that

$$(\forall \boldsymbol{x} \in \mathcal{D})(\forall l \in \mathbb{N}) \quad h(\boldsymbol{x}, l) = \operatorname{prox}_{\overline{\gamma}\theta^l f}^{\boldsymbol{M}} \left(\boldsymbol{x} - \overline{\gamma}\theta^l \boldsymbol{M}^{-1} \nabla \varphi_\mu(\boldsymbol{x}) \right).$$

Note that, from Proposition 2.2.11, for every $\boldsymbol{x} \in \mathcal{D}$ and $l \in \mathbb{N}$,

$$\boldsymbol{M}(\boldsymbol{x} - h(\boldsymbol{x}, l)) - \overline{\gamma}\theta^l \nabla \varphi_\mu(\boldsymbol{x}) \in \overline{\gamma}\theta^l \partial f(h(\boldsymbol{x}, l)). \tag{3.8}$$

First, we check that, in the backtracking procedure, if the stepsize tends to zero then the expression for the next iterate in Algorithm 11 converges to the current iterate.

Lemma 3.5.3 *Under Assumption 3.2.1, for every $\theta \in]0, 1[$, $(\overline{\gamma}, \mu) \in (\mathbb{R}_+^*)^2$ and $\boldsymbol{M} \in \mathfrak{S}_m^+$, if $\boldsymbol{x} \in \mathcal{D}$, then*

$$\lim_{l \to +\infty} h(\boldsymbol{x}, l) = \boldsymbol{x}. \tag{3.9}$$

Proof. Let $l \in \mathbb{N}$ and $\boldsymbol{x} \in \mathcal{D}$. From Proposition 2.2.10 we have

$$\|h(\boldsymbol{x}, l) - \operatorname{prox}^{\boldsymbol{M}}_{\overline{\gamma}\theta^l f}(\boldsymbol{x})\|_{\boldsymbol{M}} \leq \overline{\gamma}\theta^l \|\boldsymbol{M}^{-1}\nabla\varphi_\mu(\boldsymbol{x})\|_{\boldsymbol{M}}. \tag{3.10}$$

Taking the limit in (3.10) we deduce that,

$$\lim_{l\to+\infty} \|h(\boldsymbol{x}, l) - \operatorname{prox}^{\boldsymbol{M}}_{\overline{\gamma}\theta^l f}(\boldsymbol{x})\|_{\boldsymbol{M}} = 0 \tag{3.11}$$

In addition, $\boldsymbol{M}^{-1}\partial f$ is a maximally monotone operator with respect to $\|\cdot\|_{\boldsymbol{M}}$. From [Bauschke and Combettes, 2017, Proposition 16.27] and since $\varnothing \neq \operatorname{intdom}(f) \subset \operatorname{dom}(\partial f)$, we have

$$\mathcal{D} \subset \mathcal{C} \subset \operatorname{dom}(f) \subset \overline{\operatorname{dom}(f)} = \overline{\operatorname{intdom}(f)} \subset \overline{\operatorname{dom}(\partial f)}.$$

Thus, [Bauschke and Combettes, 2017, Theorem 23.48] leads to

$$\lim_{l\to+\infty} \|\operatorname{prox}^{\boldsymbol{M}}_{\overline{\gamma}\theta^l f}(\boldsymbol{x}) - \boldsymbol{x}\|_{\boldsymbol{M}} = 0. \tag{3.12}$$

Finally, from (3.11) and (3.12) and the triangular inequality it follows that $\lim_{l\to+\infty} \|h(\boldsymbol{x}, l) - \boldsymbol{x}\|_{\boldsymbol{M}} = 0$, hence the result since $\boldsymbol{M} \in \mathfrak{S}^+_m$.
□

We now show that, [Salzo, 2017, Lemma 3.6(ii)] holds for the line search.

Lemma 3.5.4 *Under Assumption 3.2.1, for every* $\theta \in]0, 1[$, $(\overline{\gamma}, \mu) \in (\mathbb{R}^*_+)^2$ *and* $\boldsymbol{M} \in \mathfrak{S}^+_m$, *if* $\boldsymbol{x} \in \mathcal{D} \setminus \operatorname{Argmin}(f + \varphi_\mu)$, *then*

$$\lim_{l\to+\infty} \frac{\overline{\gamma}\theta^l(\varphi_\mu(h(\boldsymbol{x}, l)) - \varphi_\mu(\boldsymbol{x}) - \langle h(\boldsymbol{x}, l) - \boldsymbol{x}, \nabla\varphi_\mu(\boldsymbol{x})\rangle)}{\|h(\boldsymbol{x}, l) - \boldsymbol{x}\|^2_{\boldsymbol{M}}} = 0.$$

Proof. Take $l \in \mathbb{N}$ and $\boldsymbol{x} \in \mathcal{D} \setminus \operatorname{Argmin}(f + \varphi_\mu)$. If $\boldsymbol{x} = h(\boldsymbol{x}, l)$ then, from (3.8), it follows that $0 \in \partial f(\boldsymbol{x}) + \nabla\varphi_\mu(\boldsymbol{x})$ and $\boldsymbol{x}$ is a minimizer of $f + \varphi_\mu$, which leads to a contradiction. Hence, $\|h(\boldsymbol{x}, l) - \boldsymbol{x}\|_{\boldsymbol{M}} \neq 0$. Since $\boldsymbol{M}^{-1}\partial f$ is a maximally monotone operator with respect to $\|\cdot\|_{\boldsymbol{M}}$, [Huang and Dong, 2014, Lemma 1] leads to

$$\frac{\|\boldsymbol{x} - h(\boldsymbol{x}, l)\|_{\boldsymbol{M}}}{\overline{\gamma}\theta^l} \leq \frac{\|\boldsymbol{x} - h(\boldsymbol{x}, l+1)\|_{\boldsymbol{M}}}{\overline{\gamma}\theta^{l+1}}$$

and $(\overline{\gamma}\theta^l/\|h(\boldsymbol{x}, l) - \boldsymbol{x}\|_{\boldsymbol{M}})_{l\in\mathbb{N}}$ is a decreasing sequence. Hence, there exists $l_0 \in \mathbb{N}$ and $a \in \mathbb{R}^*_+$ such that

$$(\forall l > l_0) \quad \frac{\overline{\gamma}\theta^l}{\|\boldsymbol{x} - h(\boldsymbol{x}, l)\|_{\boldsymbol{M}}} \leq a. \tag{3.13}$$

In addition, from Lemma 3.5.3 we deduce that (3.9) holds. According to Assumption 3.2.1(iii), $\mathcal{D}$ is an open set, so there exist $l_1 \geq l_0$ and a convex

subset $\mathcal{K}$ of $\mathcal{D}$ such that $\boldsymbol{x} \in \mathcal{K}$ and for every $l \geq l_1$, $h(\boldsymbol{x}, l) \in \mathcal{K}$. From Lemma 3.5.2 it follows that $\nabla\varphi_\mu$ is uniformly continuous on any compact subset of $\mathcal{K}$. Thus, [Salzo, 2017, Corollary 3.4 (ii)] and the norm equivalence $\lambda_{\min}(\boldsymbol{M})^{1/2}\|\cdot\| \leq \|\cdot\|_{\boldsymbol{M}} \leq \lambda_{\max}(\boldsymbol{M})^{1/2}\|\cdot\|$, where $\lambda_{\min}(\boldsymbol{M})$ and $\lambda_{\max}(\boldsymbol{M})$ are the minimal and maximal eigen values of $\boldsymbol{M}$, lead to

$$\lim_{l\to+\infty} \frac{\varphi_\mu(h(\boldsymbol{x},l)) - \varphi_\mu(\boldsymbol{x}) - \langle h(\boldsymbol{x},l) - \boldsymbol{x}, \nabla\varphi_\mu(\boldsymbol{x})\rangle}{\|h(\boldsymbol{x},l) - \boldsymbol{x}\|_{\boldsymbol{M}}} = 0. \tag{3.14}$$

Combining (3.13) and (3.14) completes the proof.
□

Finally, we derive Corollary 3.5.5 below which states that the line search performed at each iteration of Algorithm 11 is properly defined.

Corollary 3.5.5 *Let $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0, \overline{\nu}]$, and $\boldsymbol{x}_0 \in \mathcal{D}$. Suppose that Assumption 3.2.1 holds and that Algorithm 11 is run at iteration $k \in \mathbb{N}$. Then, $\boldsymbol{x}_{k+1} \in \mathcal{D}$ and condition (3.3) is met for some $l \in \mathbb{N}$.*

Proof. Let us prove the result by induction. First, note that $\boldsymbol{x}_0 \in \mathcal{D}$. Assume that the property is satisfied at iteration $k-1$ if $k \geq 1$. Under Assumption 3.2.1, if $\boldsymbol{x}_k \in \mathcal{D}$ is not a minimizer of $f + \varphi_\mu$ then we can apply Lemma 3.5.4, which implies that the line search will stop: for any $\delta > 0$, (3.3) will be satisfied for a finite l. If $\boldsymbol{x}_k \in \operatorname{Argmin}(f + \varphi_\mu)$, then $\tilde{\boldsymbol{x}}_{k,0} = \boldsymbol{x}_k$ and the line search is satisfied for $l = 0$. Moreover, if $\boldsymbol{x}_{k+1} \notin \mathcal{D}$ then $\varphi_\mu(\boldsymbol{x}_{k+1}) = +\infty$ and the inequality (3.3) could not be satisfied. Hence, $\boldsymbol{x}_{k+1} \in \mathcal{D}$.
□

3.5.1.3 Key elements for the proof of Theorem 3.4.1

Let us first show the following sufficient decrease property regarding the values of the merit function.

Lemma 3.5.6 (Sufficient decrease) *For every $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0, \overline{\nu}]$, and $\boldsymbol{x}_0 \in \mathcal{D}$, if Assumption 3.2.1 holds, then the sequence $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ produced by Algorithm 11 with $\epsilon = 0$ satisfies, for every $k \in \mathbb{N}$, the inequality*

$$\Psi_\mu(\boldsymbol{x}_{k+1}) \leq \Psi_\mu(\boldsymbol{x}_k) - \frac{\underline{\nu}(1-\delta)}{\overline{\gamma}}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2. \tag{3.15}$$

Proof. Let $k \in \mathbb{N}$. The stopping criterion (3.3) for the backtracking procedure on γ_k leads to

$$\Psi_\mu(\boldsymbol{x}_{k+1}) \leq \varphi_\mu(\boldsymbol{x}_k) + \langle \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \nabla\varphi_\mu(\boldsymbol{x}_k)\rangle + \frac{\delta}{\gamma_k}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2_{\boldsymbol{M}_k} + f(\boldsymbol{x}_{k+1}). \quad (3.16)$$

In addition, we have

$$\boldsymbol{M}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) - \gamma_k\nabla\varphi_\mu(\boldsymbol{x}_k) \in \gamma_k\partial f(\boldsymbol{x}_{k+1}),$$

and it follows from the definition of the subdifferential that

$$\gamma_k f(\boldsymbol{x}_k) \geq \langle \boldsymbol{M}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) - \gamma_k\nabla\varphi_\mu(\boldsymbol{x}_k), \boldsymbol{x}_k - \boldsymbol{x}_{k+1}\rangle + \gamma_k f(\boldsymbol{x}_{k+1}). \quad (3.17)$$

Re-writing (3.17) in a more convenient form yields

$$\langle \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \nabla\varphi_\mu(\boldsymbol{x}_k)\rangle + f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) - \frac{1}{\gamma_k}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|^2_{\boldsymbol{M}_k}. \quad (3.18)$$

Plugging (3.18) into (3.16) and using $\underline{\nu}\boldsymbol{I}_n \preceq \boldsymbol{M}_k$ completes the proof. □

Remark 3.5.7 It is worth noting that, without the assumption of existence of bounds $(\underline{\nu}, \overline{\nu})$ on matrices $(\boldsymbol{M}_k)_{k\in\mathbb{N}}$, the proof of Lemma 3.5.6 still allows us to conclude that the sequence $(\Psi_\mu(\boldsymbol{x}_k))_{k\in\mathbb{N}}$ is decreasing. Thus, in view of Theorem 2.2.21, there exists a compact $\mathcal{K} \subset \mathcal{D}$ such that $(\forall k \in \mathbb{N})\ \boldsymbol{x}_k \in \mathcal{K}$. Therefore, by the continuity of function c on $\mathcal{K}$ we deduce that there exist $\underline{c}$ and $\overline{c}$ in $]-\infty, 0[$ such that $(\forall k \in \mathbb{N})(\forall i \in \{1, \ldots, p\})\ \underline{c} \leq c_i(\boldsymbol{x}_k) \leq \overline{c} < 0$. This remark will be useful in Sections 3.6 and 3.7 to prove that the chosen variable metrics satisfy the boundedness condition.

Before deriving a lowerbound for $(\gamma_k)_{k\in\mathbb{N}}$ in Lemma 3.5.9, we show that the distance between two iterates produced by Algorithm 11 tends to zero and that the iterates are bounded.

Corollary 3.5.8 *Under Assumption 3.2.1, for every* $(\delta, \theta) \in]0, 1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}^*_+)^3$, $\underline{\nu} \in]0, \overline{\nu}]$, *and* $\boldsymbol{x}_0 \in \mathcal{D}$, *the sequence* $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ *produced by Algorithm 11 with* $\epsilon = 0$ *satisfies the following properties:*

(i) $\lim_{k\to+\infty} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| = 0$;

(ii) there exists a compact $\mathcal{K} \subset \mathcal{D}$ *such that* $(\forall k \in \mathbb{N})\ \boldsymbol{x}_k \in \mathcal{K}$.

Proof. (i) Summing (3.15) for $k = 0$ to $N - 1 \geq 0$ gives

$$\sum_{k=0}^{N-1} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 \leq \frac{\overline{\gamma}}{\underline{\nu}(1-\delta)} \left(\Psi_\mu(\boldsymbol{x}_0) - \Psi_\mu(\boldsymbol{x}_N)\right) \tag{3.19}$$

$$\leq \frac{\overline{\gamma}}{\underline{\nu}(1-\delta)} \left(\Psi_\mu(\boldsymbol{x}_0) - \underline{\Psi_\mu}\right), \tag{3.20}$$

where $\underline{\Psi_\mu} = \min_{\boldsymbol{x} \in \mathbb{R}^m} \Psi_\mu(\boldsymbol{x})$. The existence of $\underline{\Psi_\mu}$ is ensured by Corollary 3.5.1. Letting N tend to infinity gives $\sum_{k=0}^{\infty} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2 < +\infty$, which leads directly to the result.
(ii) From Theorem 2.2.21, the set $\{\boldsymbol{x} \in \mathcal{D} \mid \Psi_\mu(\boldsymbol{x}) \leq \Psi_\mu(\boldsymbol{x}_0)\}$ is compact, and from Lemma 3.5.6, for every $k \in \mathbb{N}$, $\boldsymbol{x}_k$ belongs to this set, which completes the proof.
□

Before deriving relative error and continuity conditions, we show that, for every barrier problem, the stepsize computed with the line search is bounded below from zero.

Lemma 3.5.9 *Under Assumption 3.2.1, for every* $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0, \overline{\nu}]$, *and* $\boldsymbol{x}_0 \in \mathcal{D}$, *there exists* $\underline{\gamma}_\mu > 0$ *such that the sequence* $(\gamma_k)_{k \in \mathbb{N}}$ *generated by Algorithm 11 with* $\epsilon = 0$ *is bounded below by* $\underline{\gamma}_\mu$.

Proof. Let $\mathcal{I}$ be the set of iterations in Algorithm 11 for which the stepsize value produced by the backtracking is strictly smaller than $\overline{\gamma}$, i.e. $\mathcal{I} = \{k \in \mathbb{N} \mid \gamma_k < \overline{\gamma}\}$. For every $k \in \mathcal{I}$ there exists an integer $l_k > 0$ such that $\gamma_k = \overline{\gamma}\theta^{l_k}$. By applying [Huang and Dong, 2014, Lemma 1], we have

$$(\forall k \in \mathcal{I}) \quad \frac{\|\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_{k,l_k-1}\|_{\boldsymbol{M}_k}}{\overline{\gamma}\theta^{l_k-1}} \leq \frac{\|\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_{k,l_k}\|_{\boldsymbol{M}_k}}{\overline{\gamma}\theta^{l_k}},$$

which leads to

$$(\forall k \in \mathcal{I}) \quad \|\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_{k,l_k-1}\| \leq \frac{1}{\theta} \left(\frac{\overline{\nu}}{\underline{\nu}}\right)^{\frac{1}{2}} \|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\|. \tag{3.21}$$

From Corollary 3.5.8(ii), there exists a compact subset $\mathcal{K}$ of $\mathcal{D}$ such that, for every $k \in \mathbb{N}$, $\boldsymbol{x}_k \in \mathcal{K}$. Let $\vartheta \colon \mathcal{K} \to [0, +\infty[\colon x \mapsto \operatorname{dist}(\boldsymbol{x}, \mathbb{R}^m \setminus \mathcal{D})$. ϑ is a continuous function defined on a compact set and, since $\mathcal{D}$ is open, it is positive valued. It follows from the extreme value theorem, that there exists $\eta \in]0, +\infty[$ such that $\eta = \min_{\boldsymbol{x} \in \mathcal{K}} \vartheta(\boldsymbol{x})$. For every $\boldsymbol{z} \in \mathcal{K}$, let $B(\boldsymbol{z}, \eta/2)$ be the open ball with center $\boldsymbol{z}$ and radius $\eta/2$. For every $\boldsymbol{y} \in B(\boldsymbol{z}, \eta/2)$,

$\|\boldsymbol{y}-\boldsymbol{z}\| \leq \eta/2 < \eta \leq \operatorname{dist}(\boldsymbol{z}, \mathbb{R}^m \setminus \mathcal{D})$, which implies that $\boldsymbol{y} \notin \mathbb{R}^m \setminus \mathcal{D}$, that is $\boldsymbol{y} \in \mathcal{D}$. This shows that

$$(\forall \boldsymbol{z} \in \mathcal{K}) \quad \overline{B(\boldsymbol{z}, \eta/2)} \subset \mathcal{D}. \tag{3.22}$$

On the other hand, since $\cup_{\boldsymbol{z}\in\mathcal{K}} B(\boldsymbol{z}, \eta/4)$ is a cover of $\mathcal{K}$, it follows from the compactness of this latter set that there exists $(\boldsymbol{z}_j)_{1\leq j\leq J}$ in $\mathcal{K}$ such that

$$\mathcal{K} \subset \bigcup_{j=1}^{J} B(\boldsymbol{z}_j, \eta/4). \tag{3.23}$$

Let $\mathcal{S} = \cup_{j=1}^{J} \overline{B(\boldsymbol{z}_j, \eta/2)}$. It follows from (3.22) and (3.23) that this set is a compact subset of $\mathcal{D}$ including $\mathcal{K}$. From (3.23), for every $k \in \mathcal{I}$, there exists $j_k \in \{1, \ldots, J\}$ such that

$$\|\boldsymbol{x}_k - \boldsymbol{z}_{j_k}\| < \frac{\eta}{4}. \tag{3.24}$$

On the other hand, according to Corollary 3.5.8(i), there exists $k_0 \in \mathbb{N}$ such that

$$(\forall k \geq k_0) \quad \|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\| \leq \theta \left(\frac{\nu}{\overline{\nu}}\right)^{\frac{1}{2}} \frac{\eta}{4}. \tag{3.25}$$

Set $\mathcal{I}_0 = \{k \in \mathcal{I} \mid k \geq k_0\}$. By applying the triangle inequality, we deduce from (3.21), (3.24), and (3.25) that

$$(\forall k \in \mathcal{I}_0) \quad \|\tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{z}_{j_k}\| < \frac{\eta}{2},$$

which shows that $\tilde{\boldsymbol{x}}_{k,l_k-1} \in \mathcal{S}$. Since φ_μ is convex, the following inequality holds for every $k \in \mathcal{I}_0$:

$$\begin{aligned}\langle \tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{x}_k, \nabla\varphi_\mu(\tilde{\boldsymbol{x}}_{k,l_k-1}) - \nabla\varphi_\mu(\boldsymbol{x}_k)\rangle \geq \\ \varphi_\mu(\tilde{\boldsymbol{x}}_{k,l_k-1}) - \varphi_\mu(\boldsymbol{x}_k) - \langle \tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{x}_k, \nabla\varphi_\mu(\boldsymbol{x}_k)\rangle.\end{aligned} \tag{3.26}$$

In addition, l_k is the smallest integer such that (3.3) is satisfied. Hence, (3.3) is not satisfied for $\tilde{\boldsymbol{x}}_{k,l_k-1}$ and, for every $k \in \mathcal{I}_0$, the following holds,

$$\varphi_\mu(\tilde{\boldsymbol{x}}_{k,l_k-1}) - \varphi_\mu(\boldsymbol{x}_k) - \langle \tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{x}_k, \nabla\varphi_\mu(\boldsymbol{x}_k)\rangle > \frac{\underline{\nu}\theta\delta}{\gamma_k} \|\tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{x}_k\|^2. \tag{3.27}$$

Necessarily, $\boldsymbol{x}_k \neq \tilde{\boldsymbol{x}}_{k,l_k-1}$. From (3.26) and (3.27), it follows that $(\forall k \in \mathcal{I}_0)$,

$$\|\tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{x}_k\| \, \|\nabla\varphi_\mu(\tilde{\boldsymbol{x}}_{k,l_k-1}) - \nabla\varphi_\mu(\boldsymbol{x}_k)\| > \frac{\underline{\nu}\theta\delta}{\gamma_k} \|\tilde{\boldsymbol{x}}_{k,l_k-1} - \boldsymbol{x}_k\|^2.$$

Moreover, according to Lemma 3.5.2, $\nabla\varphi_\mu$ is Lipschitz continuous on $\mathcal{S}$. Hence, there exists $L_{\mathcal{S}} \in \mathbb{R}_+^*$ such that

$$(\forall k \in \mathcal{I}_0) \quad \gamma_k > \frac{\underline{\nu}\theta\delta}{L_{\mathcal{S}}}.$$

In addition, $(\forall k \notin \mathcal{I})\ \gamma_k = \overline{\gamma}$, and the set $\mathcal{I} \setminus \mathcal{I}_0$ has a finite number of elements. Hence, the proof is complete by setting $\underline{\gamma}_\mu = \min\left\{\overline{\gamma}, \frac{\nu\theta\delta}{L_{\mathcal{S}}}, (\gamma_k)_{k\in\mathcal{I}\setminus\mathcal{I}_0}\right\}$.
□

We are now ready to identify a sequence of subgradients of Ψ_μ converging to zero.

Lemma 3.5.10 (Relative error condition) *Under Assumption 3.2.1, for every $(\delta,\theta) \in]0,1[^2$, $(\overline{\gamma},\mu,\overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0,\overline{\nu}]$, and $\boldsymbol{x}_0 \in \mathcal{D}$, the sequence $(\boldsymbol{v}_{k+1})_{k\in\mathbb{N}}$ produced by Algorithm 11 when $\epsilon = 0$ is such that $(\forall k \in \mathbb{N})$ $\boldsymbol{v}_{k+1} \in \partial\Psi_\mu(\boldsymbol{x}_{k+1})$ and*

$$(\exists K_\mu > 0)(\forall k \in \mathbb{N}) \quad \|\boldsymbol{v}_{k+1}\| \leq K_\mu \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|.$$

Proof. Let $k \in \mathbb{N}$. By definition of $\boldsymbol{x}_{k+1}$ and $\boldsymbol{v}_{k+1}$,

$$\boldsymbol{v}_{k+1} - \nabla\varphi_\mu(\boldsymbol{x}_{k+1}) = \frac{1}{\gamma_k}\boldsymbol{M}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k+1}) - \nabla\varphi_\mu(\boldsymbol{x}_k) \in \partial f(\boldsymbol{x}_{k+1}). \quad (3.28)$$

By definition of Ψ_μ, it follows that $\boldsymbol{v}_{k+1} \in \partial\Psi_\mu(\boldsymbol{x}_{k+1})$. In addition, the triangle inequality and Lemma 3.5.9 lead to

$$\|\boldsymbol{v}_{k+1}\| \leq \frac{\overline{\nu}}{\underline{\gamma}_\mu}\|\boldsymbol{x}_k - \boldsymbol{x}_{k+1}\| + \|\nabla\varphi_\mu(\boldsymbol{x}_{k+1}) - \nabla\varphi_\mu(\boldsymbol{x}_k)\|.$$

From Corollary 3.5.8(ii) and Lemma 3.5.2, we know that $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ belongs to a compact subset $\mathcal{K}$ of $\mathcal{D}$, on which $\nabla\varphi_\mu$ is Lipschitz-continuous for some constant $L_{\mathcal{K}} > 0$. Setting $K_\mu = \overline{\nu}/\underline{\gamma}_\mu + L_{\mathcal{K}}$ completes the proof.
□

Remark 3.5.11 It can also be deduced from (3.28) that, for every $k \in \mathbb{N}$, $\boldsymbol{v}_{k+1} - \mu\nabla\mathcal{B}(\boldsymbol{x}_{k+1}) \in \partial(f+g)(\boldsymbol{x}_{k+1})$.

3.5.1.4 Proof of Theorem 3.4.1

In view of Lemma 3.5.10 and Corollary 3.5.8(i), for every $\epsilon \in \mathbb{R}_+^*$, there exists $k \in \mathbb{N}$ such that $\|\boldsymbol{v}_{k+1}\| < \epsilon$.
□

3.5.2 Convergence analysis of Algorithm 11

3.5.2.1 Preliminary results

We first derive the following continuity condition.

Lemma 3.5.12 (Continuity condition) *Under Assumptions 3.2.1, let* $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0, \overline{\nu}]$ *and* $\boldsymbol{x}_0 \in \mathcal{D}$. *If the sequence* $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ *is produced by Algorithm 11 with* $\epsilon = 0$, *then there exists a subsequence* $(\boldsymbol{x}_{k_q})_{q\in\mathbb{N}}$ *and there exists* $\overline{\boldsymbol{x}} \in \mathcal{D}$ *such that*

$$\lim_{q\to+\infty} \boldsymbol{x}_{k_q} = \overline{\boldsymbol{x}} \quad \textit{and} \quad \lim_{q\to+\infty} \Psi_\mu(\boldsymbol{x}_{k_q}) = \Psi_\mu(\overline{\boldsymbol{x}}). \tag{3.29}$$

Proof. From Corollary 3.5.8(ii) we know that $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ belongs to a compact subset $\mathcal{K}$ of $\mathcal{D}$. Hence, there exists a subsequence $(\boldsymbol{x}_{k_q})_{q\in\mathbb{N}}$ converging to an element $\overline{\boldsymbol{x}} \in \mathcal{D}$. Since Ψ_μ is l.s.c., we have

$$\Psi_\mu(\overline{\boldsymbol{x}}) \leq \liminf_{q\to+\infty} \Psi_\mu(\boldsymbol{x}_{k_q}). \tag{3.30}$$

Without loss of generality one can assume that $k_0 > 0$. From Lemma 3.5.10, for every $q \in \mathbb{N}$, $\boldsymbol{v}_{k_q}$ belongs to $\partial\Psi_\mu(\boldsymbol{x}_{k_q})$ and $\|\boldsymbol{v}_{k_q}\| \leq K_\mu \|\boldsymbol{x}_{k_q} - \boldsymbol{x}_{k_q-1}\|$. For every $q \in \mathbb{N}$, we have

$$\begin{aligned} \Psi_\mu(\boldsymbol{x}_{k_q}) &\leq -\left\langle \boldsymbol{v}_{k_q}, \overline{\boldsymbol{x}} - \boldsymbol{x}_{k_q} \right\rangle + \Psi_\mu(\overline{\boldsymbol{x}}) \\ &\leq K_\mu \|\boldsymbol{x}_{k_q} - \boldsymbol{x}_{k_q-1}\| \|\overline{\boldsymbol{x}} - \boldsymbol{x}_{k_q}\| + \Psi_\mu(\overline{\boldsymbol{x}}). \end{aligned} \tag{3.31}$$

From Corollary 3.5.8(i), $\|\boldsymbol{x}_{k_q} - \boldsymbol{x}_{k_q-1}\| \to 0$ as $q \to +\infty$. Hence, taking the limit in (3.31) yields the following inequality

$$\limsup_{q\to+\infty} \Psi_\mu(\boldsymbol{x}_{k_q}) \leq \Psi_\mu(\overline{\boldsymbol{x}}). \tag{3.32}$$

Altogether (3.30) and (3.32) lead to $\lim_{q\to+\infty} \Psi_\mu(\boldsymbol{x}_{k_q}) = \Psi_\mu(\overline{\boldsymbol{x}})$.

□

The next result guarantees that, in the absence of stopping rule, Algorithm 11 converges to a solution to the barrier problem.

Proposition 3.5.13 *Let* $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \mu, \overline{\nu}) \in (\mathbb{R}_+^*)^3$, $\underline{\nu} \in]0, \overline{\nu}]$, *and* $\boldsymbol{x}_0 \in \mathcal{D}$. *Suppose that* Ψ_μ *is a KL function and that Assumption 3.2.1 holds. Then the sequence* $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ *produced by Algorithm 11 with* $\epsilon = 0$ *converges to a solution to* $\mathcal{P}_\mu$ *and has a finite length, i.e.*

$$\sum_{k=0}^{+\infty} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| < +\infty.$$

Proof. Under Assumption 3.2.1, Lemmas 3.5.6, 3.5.10 and 3.5.12 hold. If, in addition, Ψ_μ is a KL function, then we can apply [Attouch et al., 2013, Theorem 2.9]. Thus, $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ converges to a critical point of Ψ_μ and has finite length. By convexity, every critical point of Ψ_μ is a global minimizer of Ψ_μ, and a solution to $\mathcal{P}_\mu$.
□

We now show that, for a useful special case, Algorithm 11 converges linearly in terms of iterate and objective function value.

Proposition 3.5.14 *Let $\mu \in \mathbb{R}_+^*$ and consider the barrier problem $\mathcal{P}_\mu$ associated to problem (3.4). Under Assumptions 3.2.1 and 3.4.2, for every $(\delta, \theta) \in]0,1[^2$, $(\overline{\gamma}, \overline{\nu}) \in (\mathbb{R}_+^*)^2$, $\underline{\nu} \in]0, \overline{\nu}]$, and $\boldsymbol{x}_0 \in \mathcal{D}$, the sequence $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ generated by Algorithm 11 with $\epsilon = 0$ converges linearly to a solution $\boldsymbol{x}^*$ to $\mathcal{P}_\mu$, and $(\Psi_\mu(\boldsymbol{x}_k))_{k\in\mathbb{N}}$ converges linearly to $\Psi(\boldsymbol{x}^*)$.*

Proof. Let $\mu \in \mathbb{R}_+^*$. Under Assumptions 3.2.1 and 3.4.2 we can apply [Li and Pong, 2018, Corollary 5.1] which states that Ψ_μ is a KL function with exponent 1/2. The convergence of $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ to a solution $\boldsymbol{x}^* \in \mathcal{D}$ to $\mathcal{P}_\mu$ is guaranteed by Proposition 3.5.13. From Lemmas 3.5.6, 3.5.10 and 3.5.12, we can apply [Frankel et al., 2015, Theorem 4(ii)]. Hence, there exist $\bar{c} \in \mathbb{R}_+^*$ and $k_0 \in \mathbb{N}$ such that for every $k \geq k_0$,

$$\Psi_\mu(\boldsymbol{x}_k) - \Psi_\mu(\boldsymbol{x}^*) = O\left(\exp(-\bar{c}K_\mu)^k\right) \tag{3.33}$$

and

$$\|\boldsymbol{x}^* - \boldsymbol{x}_k\| = O\left(\exp(-\bar{c}K_\mu/2)^k\right). \tag{3.34}$$

The linear convergence properties follow from the fact that $\exp(-\bar{c}K_\mu) < 1$ and $\exp(-\bar{c}K_\mu/2) < 1$.
□

3.5.2.2 Proof of Theorem 3.4.4

Proposition 3.5.13 ensures the convergence of $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ to some $\boldsymbol{x}^*$. According to Lemma 3.5.10, there exists $K_\mu > 0$ such that

$$\begin{aligned}(\forall k \in \mathbb{N}) \quad \|\boldsymbol{v}_{k+1}\| &\leq K_\mu \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| \\ &\leq K_\mu(\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\| + \|\boldsymbol{x}^* - \boldsymbol{x}_k\|).\end{aligned}$$

We then deduce from Proposition 3.5.14 that there exists $\bar{c} \in \mathbb{R}_+^*$ such that $\|\boldsymbol{v}_{k+1}\| = O\left(\exp(-\bar{c}K_\mu/2)^k\right)$.
□

3.5.3 Convergence analysis of Algorithm 12

We are now ready to establish the convergence of Algorithm 12 to a solution to problem $\mathcal{P}_0$ described in (3.1). Under Assumption 3.2.1, finding a solution to $\mathcal{P}_0$ is equivalent to finding a saddle point of the associated Lagrangian $\mathcal{L}_0$, which is defined, for every $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{\lambda} \in [0, +\infty[^p$, as

$$\mathcal{L}_0(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + g(\boldsymbol{x}) + \boldsymbol{\lambda}^\top c(\boldsymbol{x}). \tag{3.35}$$

For every $\boldsymbol{x} \in \mathcal{C}$, let $\nabla c(\boldsymbol{x})$ be the matrix in $\mathbb{R}^{m \times p}$ such that $(\forall i \in \{1, \dots, p\})$ its ith column is equal to $\nabla c_i(\boldsymbol{x})$. For every $\boldsymbol{v} \in \mathbb{R}^p$ let $\mathrm{Diag}(\boldsymbol{v})$ denotes the diagonal matrix whose elements are given by $\boldsymbol{v}$. A point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ is a saddle point of $\mathcal{L}_0$ if and only if the following optimality conditions are satisfied.

$$\begin{cases} 0 \in \partial f(\boldsymbol{x}^*) + \nabla g(\boldsymbol{x}^*) + \nabla c(\boldsymbol{x}^*)\boldsymbol{\lambda}^* & (3.36) \\ \mathrm{Diag}(\boldsymbol{\lambda}^*) c(\boldsymbol{x}^*) = \mathbf{0}_p & (3.37) \\ \boldsymbol{\lambda}^* \in [0, +\infty[^p \quad \text{and} \quad c(\boldsymbol{x}^*) \in]-\infty, 0]^p & (3.38) \end{cases}$$

We prove in the following section that the sequences $(\boldsymbol{x}_j)_{j\in\mathbb{N}}$ and $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$, produced by Algorithm 12, converge to a saddle point of $\mathcal{L}_0$. We remind that, for every $j \in \mathbb{N}$, the dual variable $\boldsymbol{\lambda}_{j+1}$ is defined as follows,

$$\boldsymbol{\lambda}_{j+1} = \left(-\frac{\mu_j}{c_i(\boldsymbol{x}_{j+1})} \right)_{1 \le i \le p}. \tag{3.39}$$

As it is detailed in Section 2.2.5.2, (3.39) can be seen as a perturbation of condition (3.37) and, in the case of affine inequality constraints, (3.39) can be directly derived from the Lagrangian formulation of the barrier problem.

We first show that the primal and dual sequences produced by Algorithm 12 are bounded.

Lemma 3.5.15 *Under Assumptions 3.2.1 and 3.3.1, for every $(\delta, \theta) \in]0, 1[^2$, $(\overline{\gamma}, \mu_0) \in (\mathbb{R}_+^*)^2$, and $\boldsymbol{x}_0 \in \mathcal{D}$, the sequences $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ and $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$ produced by Algorithm 12 are bounded.*

Proof. Let $j \in \mathbb{N}$. According to Lemma 3.5.10, $\boldsymbol{v}_{j+1}$ belongs to $\partial \Psi_{\mu_j}(\boldsymbol{x}_{j+1})$. Using the definition of the subdifferential, the stopping criterion $\|\boldsymbol{v}_{j+1}\| \le \epsilon_j$, and the Cauchy-Schwarz inequality leads to

$$\begin{aligned} \Psi_{\mu_j}(\boldsymbol{x}_{j+1}) &\le \langle \boldsymbol{v}_{j+1}, \boldsymbol{x}_{j+1} - \boldsymbol{x}_0 \rangle + \Psi_{\mu_j}(\boldsymbol{x}_0) \\ &\le \epsilon_j \|\boldsymbol{x}_{j+1} - \boldsymbol{x}_0\| + \Psi_{\mu_j}(\boldsymbol{x}_0). \end{aligned} \tag{3.40}$$

Eq. (3.40) is equivalent to

$$(f+g)(\boldsymbol{x}_{j+1}) \le \mu_j(\mathcal{B}(\boldsymbol{x}_0) - \mathcal{B}(\boldsymbol{x}_{j+1})) + \epsilon_j \|\boldsymbol{x}_{j+1} - \boldsymbol{x}_0\| + (f+g)(\boldsymbol{x}_0). \tag{3.41}$$

Moreover, since $\mathcal{B}$ is convex,

$$\begin{aligned}(f+g)(\boldsymbol{x}_{j+1}) &\leq -\mu_j \langle \nabla\mathcal{B}(\boldsymbol{x}_0), \boldsymbol{x}_{j+1}-\boldsymbol{x}_0\rangle + \epsilon_j\|\boldsymbol{x}_{j+1}-\boldsymbol{x}_0\| + (f+g)(\boldsymbol{x}_0)\\ &\leq (\epsilon_j + \mu_j\|\nabla\mathcal{B}(\boldsymbol{x}_0)\|)\|\boldsymbol{x}_{j+1}-\boldsymbol{x}_0\| + (f+g)(\boldsymbol{x}_0). \qquad (3.42)\end{aligned}$$

Assume that $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ is unbounded. Then there exists a subsequence $(\boldsymbol{d}_q)_{q\in\mathbb{N}} = (\boldsymbol{x}_{j_q+1})_{q\in\mathbb{N}}$ of $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ such that the sequence $(t_q)_{q\in\mathbb{N}} = (\|\boldsymbol{x}_{j_q+1}\|)_{q\in\mathbb{N}}$ has only strictly positive elements and satisfies

$$\lim_{q\to+\infty} t_q = +\infty, \quad \lim_{q\to+\infty} \frac{\boldsymbol{d}_q}{t_q} = \overline{\boldsymbol{d}} \in \mathbb{R}^m \ \text{ and } \ \|\overline{\boldsymbol{d}}\| = 1. \qquad (3.43)$$

The last two equalities are derived from the compactness of the unit ball. Let $\boldsymbol{x}^*$ be a solution to $\mathcal{P}_0$ and let $t \in \mathbb{R}_+^*$. For every $q_0 \in \mathbb{N}$, let $\tau_{q_0} = \min\{t_q \mid q \geq q_0\}$. Since $(\forall i \in \{1,\dots,p\})\ c_i(\boldsymbol{x}^*) \leq 0$, $(\forall q \in \mathbb{N})\ c_i(\boldsymbol{d}_q) < 0$, and c_i is convex, we have

$$(\forall q \geq q_0) \quad \left(1 - \frac{\tau_{q_0}}{t_q}\right)\boldsymbol{x}^* + \frac{\tau_{q_0}}{t_q}\boldsymbol{d}_q \in \mathcal{D}.$$

By taking the limit in the above inclusion as $q \to \infty$, we obtain $\boldsymbol{x}^* + \tau_{q_0}\overline{\boldsymbol{d}} \in \mathcal{C}$. In addition, for every $q \geq q_0$,

$$(f+g)\left(\left(1 - \frac{\tau_{q_0}}{t_q}\right)\boldsymbol{x}^* + \frac{\tau_{q_0}}{t_q}\boldsymbol{d}_q\right) \leq \left(1 - \frac{\tau_{q_0}}{t_q}\right)(f+g)(\boldsymbol{x}^*) + \frac{\tau_{q_0}}{t_q}(f+g)(\boldsymbol{d}_q). \qquad (3.44)$$

We deduce from (3.42) that

$$\frac{1}{t_q}(f+g)(\boldsymbol{d}_q) \leq (\epsilon_{j_q} + \mu_{j_q}\|\nabla\mathcal{B}(\boldsymbol{x}_0)\|)\left\|\frac{\boldsymbol{d}_q}{t_q} - \frac{\boldsymbol{x}_0}{t_q}\right\| + \frac{1}{t_q}(f+g)(\boldsymbol{x}_0). \qquad (3.45)$$

As $q \to +\infty$, we have $\epsilon_{j_q} \to 0$, $\mu_{j_q} \to 0$, $t_q \to +\infty$ and $\|\boldsymbol{d}_q/t_q - \boldsymbol{x}_0/t_q\| \to 1$. Hence, taking the limit in (3.45) leads to $\limsup_{q\to\infty}(f+g)(\boldsymbol{d}_q)/t_q \leq 0$. Using now the lower-semicontinuity of $f+g$ and letting q tend to $+\infty$ in (3.44) lead to

$$\begin{aligned}(\forall q_0 \in \mathbb{N})\ (f+g)\left(\boldsymbol{x}^* + \tau_{q_0}\overline{\boldsymbol{d}}\right) &\leq \liminf_{q\to+\infty}(f+g)\left(\left(1 - \frac{\tau_{q_0}}{t_q}\right)\boldsymbol{x}^* + \frac{\tau_{q_0}}{t_q}\boldsymbol{d}_q\right)\\ &\leq (f+g)(\boldsymbol{x}^*) + \limsup_{q\to+\infty}\frac{\tau_{q_0}}{t_q}(f+g)(\boldsymbol{d}_q)\\ &\leq (f+g)(\boldsymbol{x}^*).\end{aligned}$$

Therefore, for every $q_0 \in \mathbb{N}$, $\boldsymbol{x}^* + \tau_{q_0}\overline{\boldsymbol{d}}$ is a solution to $\mathcal{P}_0$. Since $\tau_{q_0} \to +\infty$ as $q_0 \to +\infty$, the set of solution to $\mathcal{P}_0$ is unbounded. This is however in contradiction with Assumption 3.2.1(i), thus showing that $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ is bounded.

Similarly, we prove that $(\boldsymbol{\lambda}_j)_{j\in\mathbb{N}}$ is bounded. Let $j \in \mathbb{N}$ and $i \in \{1,\dots,p\}$. Since c_i is convex, $c_i(\boldsymbol{x}_{j+1}) < 0$ and $\mu_j > 0$, the following inequality holds

$$\mu_j \frac{c_i(\boldsymbol{x}_0)}{c_i(\boldsymbol{x}_{j+1})} \leq \mu_j + \frac{\mu_j}{c_i(\boldsymbol{x}_{j+1})} \left\langle \nabla c_i(\boldsymbol{x}_{j+1}), \boldsymbol{x}_0 - \boldsymbol{x}_{j+1} \right\rangle . \tag{3.46}$$

Summing (3.46) for all $i \in \{1,\dots,p\}$ leads to

$$-\left\langle c(\boldsymbol{x}_0), \boldsymbol{\lambda}_{j+1} \right\rangle \leq \mu_j p - \mu_j \left\langle \nabla \mathcal{B}(\boldsymbol{x}_{j+1}), \boldsymbol{x}_0 - \boldsymbol{x}_{j+1} \right\rangle . \tag{3.47}$$

In addition, from Remark 3.5.11 and the definition of the subdifferential of $f+g$, we deduce that

$$(f+g)(\boldsymbol{x}_{j+1}) \leq -\left\langle \boldsymbol{v}_{j+1} - \mu_j \nabla \mathcal{B}(\boldsymbol{x}_{j+1}), \boldsymbol{x}_0 - \boldsymbol{x}_{j+1} \right\rangle + (f+g)(\boldsymbol{x}_0). \tag{3.48}$$

Combining (3.47) and (3.48) yields

$$-\left\langle c(\boldsymbol{x}_0), \boldsymbol{\lambda}_{j+1} \right\rangle \leq \mu_j p + (f+g)(\boldsymbol{x}_0) - (f+g)(\boldsymbol{x}_{j+1}) + \epsilon_j \|\boldsymbol{x}_0 - \boldsymbol{x}_{j+1}\|.$$

Moreover, every component of $\boldsymbol{\lambda}_{j+1}$ and of $-c(\boldsymbol{x}_0)$ is strictly positive, hence

$$0 < -\left\langle c(\boldsymbol{x}_0), \boldsymbol{\lambda}_{j+1} \right\rangle \leq \mu_j p + (f+g)(\boldsymbol{x}_0) - (f+g)(\boldsymbol{x}^*) + \epsilon_j \|\boldsymbol{x}_0 - \boldsymbol{x}_{j+1}\|,$$

where $\boldsymbol{x}^*$ is a solution to $\mathcal{P}_0$. Since $(\mu_j, \epsilon_j) \to (0,0)$ as $j \to +\infty$, and since $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ has been shown to be bounded, we conclude that $-\left\langle c(\boldsymbol{x}_0), \boldsymbol{\lambda}_{j+1} \right\rangle$ is bounded and so is $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$.
□

3.5.3.1 Proof of Theorem 3.4.5

According to Lemma 3.5.15, the sequences $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ and $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$ are bounded. Hence, there exists a subsequence $(\boldsymbol{x}_{j_q+1}, \boldsymbol{\lambda}_{j_q+1})_{q\in\mathbb{N}}$ converging to some point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$. By construction, for every $q \in \mathbb{N}$, $c(\boldsymbol{x}_{j_q+1}) \in]-\infty, 0[^p$ and $\boldsymbol{\lambda}_{j_q+1} \in]0, +\infty[^p$. Since $(\forall i \in \{1,\dots,p\})$ c_i is l.s.c., taking the limit as $q \to +\infty$ yields (3.38). Moreover, by definition, for every $q \in \mathbb{N}$ we have $\mathrm{Diag}(\boldsymbol{\lambda}_{j_q+1}) c(\boldsymbol{x}_{j_q+1}) = -\mu_{j_q} \mathbf{1}_p$. Since $(\forall i \in \{1,\dots,p\})$ c_i is continuous on $\mathcal{C}$, letting $q \to +\infty$ in the previous equality leads to (3.37). Let $q \in \mathbb{N}$. From Remark 3.5.11, it follows that $\boldsymbol{x}_{j_q+1} = \mathrm{prox}_{f+g}(\boldsymbol{x}_{j_q+1} + \boldsymbol{v}_{j_q+1} - \mu_{j_q} \nabla \mathcal{B}(\boldsymbol{x}_{j_q+1}))$. In addition, prox_{f+g} is nonexpansive. Hence,

$$\begin{aligned} &\left\| \boldsymbol{x}_{j_q+1} - \mathrm{prox}_{f+g}\left(\boldsymbol{x}^* - \nabla c(\boldsymbol{x}^*) \boldsymbol{\lambda}^*\right) \right\| \\ &\qquad \leq \left\| \boldsymbol{x}_{j_q+1} + \boldsymbol{v}_{j_q+1} - \mu_{j_q} \nabla \mathcal{B}(\boldsymbol{x}_{j_q+1}) - \boldsymbol{x}^* + \nabla c(\boldsymbol{x}^*) \boldsymbol{\lambda}^* \right\| . \end{aligned} \tag{3.49}$$

By applying the triangle inequality, we deduce from (3.49) that

$$\left\|\boldsymbol{x}^* - \operatorname{prox}_{f+g}\left(\boldsymbol{x}^* - \nabla c(\boldsymbol{x}^*)\boldsymbol{\lambda}^*\right)\right\| \leq 2\|\boldsymbol{x}^* - \boldsymbol{x}_{j_q+1}\| + \epsilon_j + \sum_{i=1}^{p} \|(\boldsymbol{\lambda}_{j_q+1})_i \nabla c_i(\boldsymbol{x}_{j_q+1}) - \lambda_i^* \nabla c_i(\boldsymbol{x}^*)\|. \quad (3.50)$$

The sequence $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ is a bounded sequence in $\mathcal{D}$. Therefore, all its cluster points belong to a compact subset of $\mathcal{C}$. In view of Assumption 3.2.1(iii), ∇c_i is continuous at $\boldsymbol{x}^*$. Thus, taking the limit in (3.50) as $q \to +\infty$ leads to

$$\boldsymbol{x}^* = \operatorname{prox}_{f+g}\left(\boldsymbol{x}^* - \nabla c(\boldsymbol{x}^*)\boldsymbol{\lambda}^*\right), \quad (3.51)$$

which is equivalent to (3.36). Finally, $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ is a saddle-point for the Lagrangian (3.35), which completes the proof.
□

3.5.3.2 Analytic center

When there are several primal or dual solutions to the constrained problem (3.1), assumptions are needed to prove the uniqueness of the cluster point exhibited in Theorem 3.4.5. Under these assumptions, the analytic center introduced in Section 2.2.5.4 provides an insightful characterization for the limit point. We show in this section the existence of a primal and a dual analytic centers.

Let us recall that under Assumption 3.2.1, strong duality holds and the set of saddle points for the Lagrangian $\mathcal{L}_0$ is equal to $\mathcal{S}_P \times \mathcal{S}_D$ where $\mathcal{S}_P$ and $\mathcal{S}_D$ are the so-called sets of primal and dual solutions to $\mathcal{P}_0$ [Boyd and Vandenberghe, 2004, Section 5.4.2]. We derive the following result for these sets.

Proposition 3.5.16 *Under Assumptions 3.2.1(i)-(ii), $\mathcal{S}_P$ and $\mathcal{S}_D$ are non-empty bounded convex sets.*

Proof. The results for $\mathcal{S}_P$ directly follows from Assumptions 3.2.1(i)-(ii). The convexity of $\mathcal{S}_D$ follows from standard results [Boyd and Vandenberghe, 2004, Section 5.2]. The fact that $\mathcal{S}_D$ is not empty also follows from Assumption 3.2.1(i)-(ii). For every $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \in \mathcal{S}_P \times \mathcal{S}_D$, the inequality $(f+g)(\boldsymbol{x}^*) \leq (f+g)(\boldsymbol{x}_0) + \boldsymbol{\lambda}^{*\top} c(\boldsymbol{x}_0)$ holds for every $\boldsymbol{x}_0 \in \mathcal{D}$. If $\mathcal{S}_D$ is unbounded, then the right-hand side can tend to $-\infty$ which is in contradiction with $(f+g)(\boldsymbol{x}^*)$ being a finite number. So $\mathcal{S}_D$ is bounded.
□

We also show that under an additional assumption, the sets $\mathcal{J}_P$, defined in (3.5), and $\mathcal{J}_D$, defined in (3.6), are nonempty. The number of elements in $\mathcal{J}_P$ and $\mathcal{J}_D$ are noted $\#\mathcal{J}_P$ and $\#\mathcal{J}_D$, respectively.

Proposition 3.5.17 *If Assumption 3.4.6 holds and if the set $\mathcal{S}_P$ (resp. $\mathcal{S}_D$) contains at least two distinct elements, then the set $\mathcal{J}_P$ (resp. $\mathcal{J}_D$) is nonempty.*

Proof. Assume that there are at least two distinct elements $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in $\mathcal{S}_P$. Then, for every $i \in \{1, \dots, p\}$, $c_i(\boldsymbol{x}_1) \leq 0$ and $c_i(\boldsymbol{x}_2) \leq 0$. Assume that Assumption 3.4.6 holds. If the constraints are affine, i.e. $c : \boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$, with $\boldsymbol{A}$ an injective matrix, then there exists $i_0 \in \{1, \dots, p\}$ such that $(\boldsymbol{A}\boldsymbol{x}_1)_{i_0} \neq (\boldsymbol{A}\boldsymbol{x}_2)_{i_0}$ and $c_{i_0}(\boldsymbol{x}_1) \neq c_{i_0}(\boldsymbol{x}_2)$. The same conclusion obviously holds if $i_0 \in \{1, \dots, p\}$ is such that c_{i_0} is strictly convex. Hence, we have either $c_{i_0}(\boldsymbol{x}_1) < 0$ or $c_{i_0}(\boldsymbol{x}_2) < 0$, that is $i_0 \in \mathcal{J}_P$.
Assume that there are at least two distinct elements $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ in $\mathcal{S}_D$. Both $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ belong to $[0, +\infty[^p$. Since $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are distinct there exists $i_0 \in \{1, \dots, p\}$ such that $(\boldsymbol{\lambda}_1)_{(i_0)} \neq (\boldsymbol{\lambda}_2)_{(i_0)}$. Hence, we have either $(\boldsymbol{\lambda}_1)_{(i_0)} > 0$ or $(\boldsymbol{\lambda}_2)_{(i_0)} > 0$, that is $i_0 \in \mathcal{J}_D$. □

If $\mathcal{J}_P$ is nonempty, we define the following quantities: for every $\boldsymbol{x} \in \mathbb{R}^m$, $c_{\mathcal{J}_P}(\boldsymbol{x}) = (c_i(\boldsymbol{x}))_{i \in \mathcal{J}_P}$ and $\mathcal{B}_{\mathcal{J}_P}(\boldsymbol{x}) = -\sum_{i \in \mathcal{J}_P} \ln(-c_i(\boldsymbol{x}))$ if $c_{\mathcal{J}_P}(\boldsymbol{x}) \in]-\infty, 0[^{\#\mathcal{J}_P}$, $+\infty$ otherwise. We also consider the following problem.

$$\mathcal{P}_P : \quad \begin{array}{ll} \underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} & \mathcal{B}_{\mathcal{J}_P}(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \mathcal{S}_P \text{ and } c_{\mathcal{J}_P}(\boldsymbol{x}) \in]-\infty, 0[^{\#\mathcal{J}_P}. \end{array} \tag{3.52}$$

Similarly, if $\mathcal{J}_D$ is nonempty, we define the following quantities: for every $\boldsymbol{\lambda} \in \mathbb{R}^p$, $\boldsymbol{\lambda}_{\mathcal{J}_D} = (\lambda_i)_{i \in \mathcal{J}_D}$ and $b_{\mathcal{J}_D}(\boldsymbol{\lambda}) = -\sum_{i \in \mathcal{J}_D} \ln(\lambda_i)$ if $\boldsymbol{\lambda}_{\mathcal{J}_D} \in]0, +\infty[^{\#\mathcal{J}_D}$, $+\infty$ otherwise. We also consider the following problem.

$$\mathcal{P}_D : \quad \begin{array}{ll} \underset{\boldsymbol{\lambda} \in \mathbb{R}^p}{\text{minimize}} & b_{\mathcal{J}_D}(\boldsymbol{\lambda}) \\ \text{subject to} & \boldsymbol{\lambda} \in \mathcal{S}_D \text{ and } \boldsymbol{\lambda}_{\mathcal{J}_D} \in]0, +\infty[^{\#\mathcal{J}_D} \end{array} \tag{3.53}$$

Lemma 3.5.18 *Under Assumptions 3.2.1 and 3.4.6, if $\mathcal{S}_P$ (resp. $\mathcal{S}_D$) does not reduce to a singleton, then there exists a unique solution to $\mathcal{P}_P$ (resp. $\mathcal{P}_D$) called the analytic center of $\mathcal{S}_P$ (resp. $\mathcal{S}_D$).*

Proof. Assume that $\mathcal{S}_P$ does not reduce to a singleton. According to Proposition 3.5.16, $\mathcal{S}_P$ is nonempty and it thus contains at least two distinct elements. It then follows from Proposition 3.5.17 that the set $\mathcal{J}_P$ is nonempty. In addition, by invoking again Proposition 3.5.16, $\mathcal{S}_P$ is bounded and convex. Since the constraint functions $(c_i)_{1 \leq i \leq p}$ are convex, we deduce that the

feasible set of $\mathcal{P}_P$ is bounded and convex. Because of the convexity of the set $\mathcal{S}_P$ and the functions $(c_i)_{1\leq i\leq p}$, it can be checked that this feasible set is nonempty. In addition, under Assumptions 3.2.1 and 3.4.6, $\mathcal{B}_{\mathcal{J}_P}$ is l.s.c. and strictly convex, and it is finite-valued on $\{\boldsymbol{x}\in\mathbb{R}^m \mid c_{\mathcal{J}_P}(\boldsymbol{x})\in]-\infty,0[^{\#\mathcal{J}_P}\}$. Hence, there exists a unique solution to $\mathcal{P}_P$. Assume that $\mathcal{S}_D$ does not reduce to a singleton. It then follows from Propositions 3.5.16 and 3.5.17 that $\mathcal{J}_D\neq\varnothing$. By using a similar reasoning as for $\mathcal{P}_P$ we deduce that there exists a unique solution to $\mathcal{P}_D$.
□

The complementary slackness property in (3.37) ensures that $\mathcal{J}_P\cap J_D=\varnothing$. We say that $\mathcal{P}_0$ has the *strict complementarity property* if $\mathcal{J}_P\cup\mathcal{J}_D=\{1,\ldots,p\}$. It follows from [Bonnans and Shapiro, 2013, Theorem 3.133] that strict complementarity always holds in linear programming. We refer to Remark 2.2.24 for a discussion regarding this assumption.

3.5.3.3 Proof of Theorem 3.4.7

(i) The result follows from Theorem 3.4.5.
(ii) Let $(\boldsymbol{x}^*,\boldsymbol{\lambda}^*)$ be a primal-dual solution to $\mathcal{P}_0$. Let $(\overline{\boldsymbol{x}},\overline{\boldsymbol{\lambda}})$ be a cluster point of $(\boldsymbol{x}_{j+1},\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$ and let $(\boldsymbol{x}_{j_q+1},\boldsymbol{\lambda}_{j_q+1})_{q\in\mathbb{N}}$ be a subsequence converging to this point. Pick $q\in\mathbb{N}$. In view of Remark 3.5.11 and (3.36) we have that

$$\boldsymbol{v}_{j_q+1}-\nabla c(\boldsymbol{x}_{j_q+1})\boldsymbol{\lambda}_{j_q+1}\in\partial(f+g)(\boldsymbol{x}_{j_q+1}),$$

and

$$-\nabla c(\boldsymbol{x}^*)\boldsymbol{\lambda}^*\in\partial(f+g)(\boldsymbol{x}^*).$$

Since $f+g$ is convex, we deduce from the monotonicity of its subdifferential that

$$0\leq\left\langle\boldsymbol{x}_{j_q+1}-\boldsymbol{x}^*,\boldsymbol{v}_{j_q+1}\right\rangle-\left\langle\boldsymbol{x}_{j_q+1}-\boldsymbol{x}^*,\nabla c(\boldsymbol{x}_{j_q+1})\boldsymbol{\lambda}_{j_q+1}-\nabla c(\boldsymbol{x}^*)\boldsymbol{\lambda}^*\right\rangle. \tag{3.54}$$

In addition, $\|\boldsymbol{v}_{j_q+1}\|\leq\epsilon_{j_q}$ and $(\forall i\in\{1,\ldots,p\})$ c_i is convex. Hence, we deduce from (3.54) that

$$0\leq\|\boldsymbol{x}_{j_q+1}-\boldsymbol{x}^*\|\epsilon_{j_q}+\big(c(\boldsymbol{x}_{j_q+1})-c(\boldsymbol{x}^*)\big)^\top\boldsymbol{\lambda}^*+\big(c(\boldsymbol{x}^*)-c(\boldsymbol{x}_{j_q+1})\big)^\top\boldsymbol{\lambda}_{j_q+1}. \tag{3.55}$$

From (3.37) and (3.39), $c(\boldsymbol{x}^*)^\top\boldsymbol{\lambda}^*=0$ and, for every $i\in\{1,\ldots,p\}$, $c_i(\boldsymbol{x}_{j_q+1})(\boldsymbol{\lambda}_{j_q+1})_i=-\mu_{j_q}$. Hence,

$$0\leq\|\boldsymbol{x}_{j_q+1}-\boldsymbol{x}^*\|\frac{\epsilon_{j_q}}{\mu_{j_q}}+p-\sum_{i=1}^{p}\frac{\lambda_i^*}{(\boldsymbol{\lambda}_{j_q+1})_i}+\frac{c_i(\boldsymbol{x}^*)}{c_i(\boldsymbol{x}_{j_q+1})}. \tag{3.56}$$

Note that $(\forall i\notin\mathcal{J}_P)$ $c_i(\boldsymbol{x}^*)=0$ and $(\forall i\notin\mathcal{J}_D)$ $\lambda_i^*=0$. If $\mathcal{J}_P$ (resp. $\mathcal{J}_D$) is nonempty, we can then choose $\boldsymbol{x}^*$ (resp. $\boldsymbol{\lambda}^*$) such that $c_{\mathcal{J}_P}(\boldsymbol{x}^*)\in]-\infty,0[^{\#\mathcal{J}_P}$

(resp. $\boldsymbol{\lambda}^*_{\mathcal{J}_D} \in]0, +\infty[^{\#\mathcal{J}_D}$). Consequently, by using Assumption 3.3.1, as $q \to +\infty$,(3.56) becomes

$$\sum_{i\in\mathcal{J}_D} \frac{\lambda_i^*}{\overline{\lambda}_i} + \sum_{i\in\mathcal{J}_P} \frac{c_i(\boldsymbol{x}^*)}{c_i(\overline{\boldsymbol{x}})} \leq p, \tag{3.57}$$

where we necessarily have

$$(\forall i \in \mathcal{J}_D) \;\; \overline{\lambda}_i > 0 \;\; \text{and} \;\; (\forall i \in \mathcal{J}_P) \;\; c_i(\overline{\boldsymbol{x}}) < 0. \tag{3.58}$$

Because of the strict complementarity, there are exactly p positive terms in the left-hand side of (3.57). Therefore, we can apply the arithmetic-geometric mean inequality which leads to

$$\left(\prod_{i\in\mathcal{J}_D} \frac{\lambda_i^*}{\overline{\lambda}_i}\right)\left(\prod_{i\in\mathcal{J}_P} \frac{c_i(\boldsymbol{x}^*)}{c_i(\overline{\boldsymbol{x}})}\right) \leq 1, \tag{3.59}$$

with the convention that, if $\mathcal{J}_P$ (resp. $\mathcal{J}_D$) is empty, the corresponding product is equal to 1. From Theorem 3.4.5, we deduce that $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{\lambda}})$ is a primal-dual solution to $\mathcal{P}_0$. Hence, $(\boldsymbol{x}^*, \overline{\boldsymbol{\lambda}})$ and $(\overline{\boldsymbol{x}}, \boldsymbol{\lambda}^*)$ are also primal-dual solutions to $\mathcal{P}_0$. Therefore, because $(\overline{\boldsymbol{x}}, \overline{\boldsymbol{\lambda}})$ satisfies (3.58), (3.59) also holds when either $\boldsymbol{\lambda}^* = \overline{\boldsymbol{\lambda}}$ or $\boldsymbol{x}^* = \overline{\boldsymbol{x}}$. Consequently,

$$\prod_{i\in\mathcal{J}_P} (-c_i(\boldsymbol{x}^*)) \leq \prod_{i\in\mathcal{J}_P} (-c_i(\overline{\boldsymbol{x}})) \;\text{and}\; \prod_{i\in\mathcal{J}_D} \lambda_i^* \leq \prod_{i\in\mathcal{J}_D} \overline{\lambda}_i. \tag{3.60}$$

If there exist at least two distinct elements in $\mathcal{S}_P$ (resp. $\mathcal{S}_D$) then, from Propositions 3.5.16 and 3.5.17, $\mathcal{J}_P$ (resp. $\mathcal{J}_D$) is nonempty. It follows from (3.60) that $\overline{\boldsymbol{x}}$ (resp. $\overline{\boldsymbol{\lambda}}$) is a solution to $\mathcal{P}_P$ (resp. $\mathcal{P}_D$). In turn, Lemma 3.5.18 guarantees that $\mathcal{P}_P$ (resp. $\mathcal{P}_D$) has a unique solution. Thus, there exists a unique cluster point for the primal (resp. dual) sequence and $(\boldsymbol{x}_{j+1})_{j\in\mathbb{N}}$ (resp. $(\boldsymbol{\lambda}_{j+1})_{j\in\mathbb{N}}$) converges to the analytic center of $\mathcal{S}_P$ (resp. $\mathcal{S}_D$).
□

Now that we have established some theoretical guarantees regarding the proposed algorithm, we show that PIPA performs well with respect to state-of-the-art methods on two applications in image processing, namely hyperspectral unmixing and joint geometry-texture decomposition and reconstruction of computed tomography data. Our numerical experiments demonstrate in addition the benefits of using a variable metric to accelerate the convergence.

§ 3.6 Application: Hyperspectral unmixing

Hyperspectral imaging devices are remote sensing systems that acquire the emitting light spectrum of a distant scene, here modeled as a 2D image [Bioucas-Dias et al., 2012]. Let s and r be respectively the number of acquired spectral bands and pixels in the image, and let $\boldsymbol{Y} \in \mathbb{R}^{s \times r}$ denote the measured hyperspectral cube. Assume that we have access to a library $\boldsymbol{S} \in \mathbb{R}^{s \times q}$, where each column of $\boldsymbol{S}$ contains the spectral signatures of one material (or endmember) among q that are expected to be present in the scene. The proportion or abundance of every material in every pixel is described through the abundance matrix $\boldsymbol{X} \in \mathbb{R}^{q \times r}$. The following linear model is frequently used to relate the data, the endmembers and the abundances, when there is no microscopic interaction between the materials:

$$\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{X} + \boldsymbol{\Omega},$$

with $\boldsymbol{\Omega} \in \mathbb{R}^{s \times r}$ a realization of an additive white Gaussian noise. The estimation of $\boldsymbol{X}$ from $\boldsymbol{Y}$ and $\boldsymbol{S}$ is an inverse problem called unmixing [Chan et al., 2018].

3.6.1 Problem formulation

Following [Iordache et al., 2012; Chouzenoux et al., 2014a], we propose to formulate the following constrained minimization problem to perform the unmixing task,

$$\begin{aligned} \underset{\boldsymbol{X} \in \mathbb{R}^{q \times r}}{\text{minimize}} \quad & \tfrac{1}{2}\|\boldsymbol{Y} - \boldsymbol{S}\boldsymbol{X}\|_2^2 + \kappa \sum_{i=1}^{q} \|(\boldsymbol{W}\boldsymbol{X}_i)_{\mathrm{d}}\|_1 \\ \text{subject to} \quad & (\forall j \in \{1, \ldots, r\}) \quad \sum_{i=1}^{q} X_{i,j} \leq 1 \\ & (\forall i \in \{1, \ldots, q\})(\forall j \in \{1, \ldots, r\}) \quad X_{i,j} \geq 0, \end{aligned} \tag{3.61}$$

where $\|\cdot\|_2$ denotes the Frobenius norm, $(\forall i \in \{1, \ldots, q\})$ $\boldsymbol{X}_i \in \mathbb{R}^r$ is the ith line of the abundance matrix $\boldsymbol{X}$, $\boldsymbol{W} \in \mathbb{R}^{r \times r}$ is a wavelet decomposition operator, $\|(\cdot)_{\mathrm{d}}\|_1$ is the ℓ_1-norm of the detail wavelet coefficients, and $\kappa \in \mathbb{R}_+$ is a regularization parameter. It is worth noting that the linear constraints account for the atmospheric absorption [Keshava and Mustard, 2002] since, for every pixel, the sum of all fractional abundances may be less than one. Moreover, the wavelet-based penalization allows us to enforce useful spatial regularity on the sought abundance maps [Pustelnik et al., 1999].

From this point forward, the vectorizations of $\boldsymbol{X} \in \mathbb{R}^{q \times r}$ and $\boldsymbol{Y} \in \mathbb{R}^{s \times r}$, in lexicographic order, are denoted by $\boldsymbol{x} \in \mathbb{R}^m$ with $m = qr$ and $\boldsymbol{y} \in \mathbb{R}^{sr}$, respectively, and $\otimes$ denotes the Kronecker product. Problem (3.61) can thus be re-written as in (3.1), with $p = m + r$ and $(\forall \boldsymbol{x} \in \mathbb{R}^m)$ $g(x) = 1/2\|\boldsymbol{y} -$

$(\boldsymbol{I}_r \otimes \boldsymbol{S})\boldsymbol{x}\|^2$, $f(\boldsymbol{x}) = \kappa \sum_{i=1}^{q} \|(\boldsymbol{W}\boldsymbol{P}_i x)_{\mathrm{d}}\|_1$, $(\forall i \in \{1,\dots,q\})$ $\boldsymbol{P}_i \in \mathbb{R}^{r\times m}$ is a decimation matrix such that $\boldsymbol{P}_i\boldsymbol{x} = \boldsymbol{X}_i$, and $c(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$ with

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{I}_r \otimes \boldsymbol{1}_q^\top \\ -\boldsymbol{I}_m \end{pmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{1}_r \\ \boldsymbol{0}_m \end{pmatrix}.$$

The resulting minimization problem satisfies Assumptions 3.2.1, 3.4.2 and 3.4.6. Hence, Theorems 3.4.1, 3.4.4 and 3.4.5 regarding convergence and convergence rate of Algorithm 11 hold. In addition, in the considered example, the rank of $\boldsymbol{S}$ is equal to q, so there exists a unique solution to (3.61) and Theorem 3.4.7(i) holds.

3.6.2 Realistic data simulation and test configuration

In order to simulate $\boldsymbol{Y}$, we make use of the Urban[1] dataset, which provides the spectral signatures and abundance maps for $q = 6$ materials in $s = 162$ spectral bands. We consider images of size $r = 256 \times 256$. The product of the spectral library and attenuated abundance map is corrupted with an additive white Gaussian noise with a standard deviation of 0.06. Let the signal-to-noise ratio be defined as

$$\mathrm{SNR} = 20\log_{10}(\|\overline{\boldsymbol{x}}\|/\|\boldsymbol{x} - \overline{\boldsymbol{x}}\|),$$

where $\overline{\boldsymbol{x}}$ is the ground-truth for $\boldsymbol{x}$. In addition, for each material $i \in \{1,\dots,q\}$, the signal-to-noise ratio of its associated abundance map $\boldsymbol{X}_i \in \mathbb{R}^r$ is

$$\mathrm{SNR}_i = 20\log_{10}\left(\|\overline{\boldsymbol{X}}_i\|/\|\boldsymbol{X}_i - \overline{\boldsymbol{X}}_i\|\right).$$

The regularization weight κ is tuned by a grid search so as to reach the largest SNR, in that case $\kappa = 0.01$. Regarding the operator $\boldsymbol{W}$, we selected an orthogonal Daubechies 4 wavelet decomposition performed over 2 resolution levels. As for the variable metric, we consider two cases: the proposed method without variable metric (taken as the identity matrix), which is referred to as PIPA; and PIPA-VM, which denotes the case when, following the strategy in [Becker and Fadili, 2012], for every $j \in \mathbb{N}$, the variable metric is chosen as the Hessian of φ_{μ_j}. The proximity operator of the regularization term in the variable metric is computed numerically using Algorithm 5.

Let us now discuss the boundedness condition required for the variable metrics in PIPA-VM. For every $\boldsymbol{x} \in \mathcal{D}$, we have

$$\varphi_\mu(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - (\boldsymbol{I}_r \otimes \boldsymbol{S})\boldsymbol{x}\|^2 - \mu \sum_{i=1}^{p} \ln(b_i - \boldsymbol{A}_i^\top \boldsymbol{x}),$$

[1] www.escience.cn/people/feiyunZHU/Dataset_GT.html

where $\boldsymbol{A}_i \in \mathbb{R}^m$ (resp. $b_i \in \mathbb{R}$) is the ith row (resp. component) of $\boldsymbol{A}$ (resp. of $\boldsymbol{b}$). For every $\mu \in \mathbb{R}_+^*$ and $\boldsymbol{x} \in \mathcal{D}$, the Hessian of φ_μ at $\boldsymbol{x}$ is equal to

$$\nabla^2 \varphi_\mu(\boldsymbol{x}) = \boldsymbol{I}_r \otimes (\boldsymbol{S}^\top \boldsymbol{S}) + \mu \sum_{i=1}^{p} \frac{\boldsymbol{A}_i \boldsymbol{A}_i^\top}{(\boldsymbol{A}_i^\top \boldsymbol{x} - b_i)^2}.$$

Finally, in view of Remark 3.5.7, we deduce that there exist $\underline{c}$ and $\overline{c}$ in $]-\infty, 0[$ such that, for every $k \in \mathbb{N}$ and every $i \in \{1, \ldots, p\}$, $\underline{c} \leq \boldsymbol{A}_i^\top \boldsymbol{x}_k - b_i \leq \overline{c} < 0$. Since the rank of $\boldsymbol{A}$ is equal to m, for every $\mu \in \mathbb{R}_+^*$, the aforementioned variable metrics are bounded from below and above by strictly positive constants, as required in Algorithm 11.

In order to satisfy Assumption 3.3.1, we choose the barrier parameter and precision sequences as follows,

$$(\forall j \in \mathbb{N}) \quad \epsilon_j = \overline{\epsilon} \frac{\mu_j}{\zeta^j}, \quad \text{and} \quad \mu_{j+1} = \frac{\mu_j}{\rho_j}, \tag{3.62}$$

where $\rho_j \geq \underline{\rho} > 1$, $\zeta > 1$ and $\overline{\epsilon} > 0$. Regarding PIPA, we take $\overline{\epsilon} = 10^3$, $\mu_0 = 1$, $\rho_j = 1.5$ for every $j \in \mathbb{N}$ and $\zeta = 1 + 10^{-5}$. For PIPA-VM we choose $\overline{\epsilon} = 10^5$, $\mu_0 = 0.01$, $\zeta = 1 + 10^{-5}$ and, to avoid numerical instabilities when μ_j is very small, we pick $\rho_j = 1.5$ for every $j \in \mathbb{N}$ such that $\mu_j \geq 10^{-6}$, and decrease it gradually: if $4 \times 10^{-9} \leq \mu_j < 10^{-6}$ then $\rho_j = 1.1$, if $10^{-12} \leq \mu_j < 4 \times 10^{-9}$ then $\rho_j = 1.01$, and finally, if $\mu_j < 10^{-12}$, then $\rho_j = 1.001$.

We compare PIPA and PIPA-VM with three state-of-the-art convex optimization algorithms: ADMM [Setzer et al., 2010; Iordache et al., 2012], which is presented in Algorithm 8, PDS [Komodakis and Pesquet, 2015; Combettes et al., 2014], summarized in Algorithm 6, and GFBS [Raguet et al., 2013], which is presented in Algorithm 4.

We also implement preconditioned versions of ADMM and GFBS, which are referred to as ADMM-VM and GFBS-VM, respectively. ADMM-VM is based on [Shefi and Teboulle, 2014, Algorithm 2], where the metrics are taken constant as in [Shefi and Teboulle, 2014, Example 3.4]. Regarding GFBS-VM, we implement [Raguet and Landrieu, 2015, Algorithm 1] with a modified metric based on the Hessian of the data-fitting term in (3.61). All computational times are given for experiments run on Matlab 2018b on an Intel Xeon CPU E5-1650 at frequency 3.20 GHz.

3.6.3 Results

The solution to (3.61) with $\kappa = 0$, i.e. without regularization, can be obtained with the primal-dual interior point method from [Chouzenoux et al., 2014a]. It is referred to as IPLS and yields SNR = 11.02 dB, whereas solving the same problem with $\kappa = 0.01$ leads to a better reconstruction with SNR = 12.45 dB, illustrating the benefits of regularizing in this example. Figure 3.1 shows that the SNR increases faster with PIPA-VM than with the

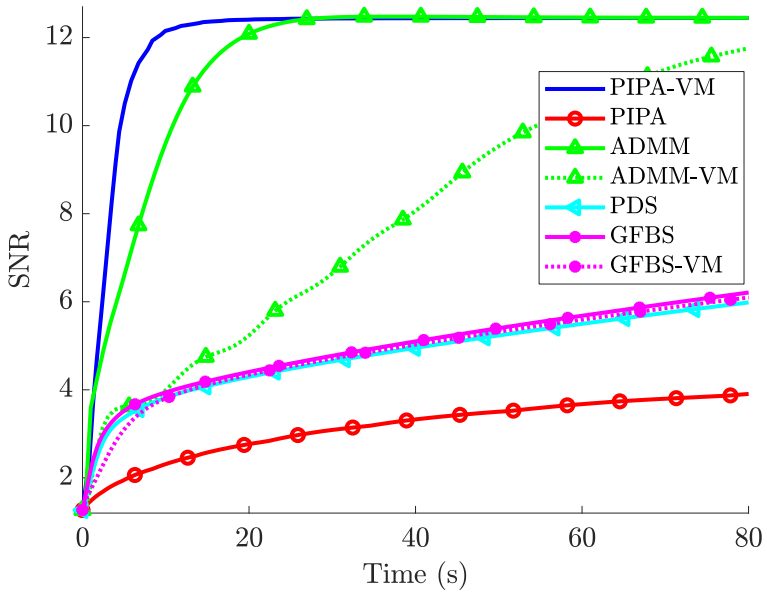


Figure 3.1: *SNR as a function of time.*

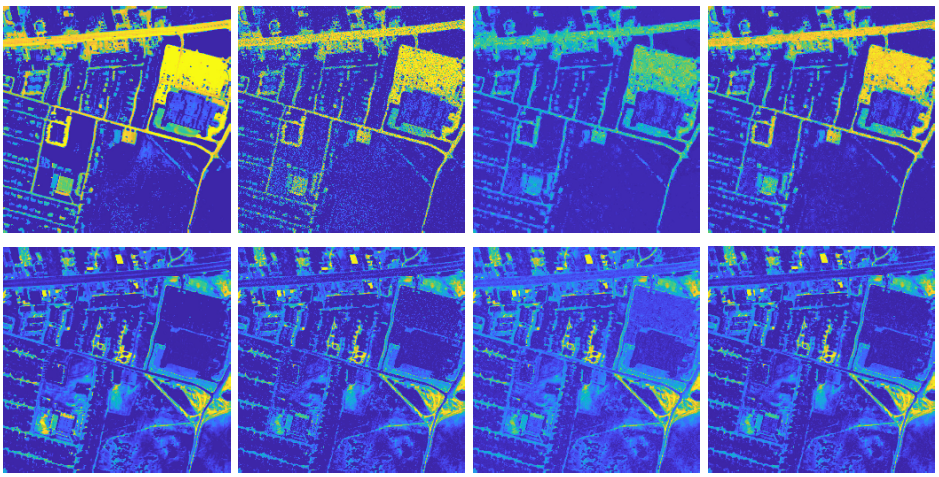
other algorithms. Moreover, it can be clearly seen in this figure that PIPA-VM exhibits a much faster convergence than PIPA, which demonstrates the advantage of using a variable metric in this example. The SNR obtained for each material after running the different methods for 11 seconds can be found in Table 3.1. For 5 out of 6 endmembers, the SNR of the abundance maps obtained with PIPA-VM after 11 seconds is better than for all other methods. In addition, for all materials PIPA-VM gives better results after 11 seconds than the non-regularized solution IPLS.

	Asphalt	Grass	Tree	Roof	Metal	Dirt
IPLS	10.12	11.21	11.86	14.91	4.90	13.68
PDS	2.23	3.65	4.83	8.56	7.12	10.30
GFBS	2.31	3.82	4.88	9.38	0.40	10.24
GFBS-VM	2.50	3.58	4.86	7.46	-0.42	10.71
ADMM	7.40	11.37	12.45	15.08	**7.25**	12.34
ADMM-VM	2.78	4.90	4.01	8.02	0.70	6.79
PIPA	1.71	2.17	2.90	2.20	-0.94	8.51
PIPA-VM	**11.31**	**12.25**	**13.04**	**15.27**	7.12	**14.52**

Table 3.1: *Signal-to-noise ratio* $(SNR_i)_{1\leq i\leq 6}$ *(in dB) for each material after 11 seconds.*

Visual results for Asphalt and Dirt materials are displayed in Figure 3.2,

where we only show the results for IPLS, ADMM and PIPA-VM, since they outperformed the other methods. One can notice that, after running all algorithms for 11 sec, the abundance maps produced by PIPA-VM for these two materials are visually more satisfactory than the ones obtained with ADMM, while the non-regularized solution IPLS is significantly noisy.



(a) (b) (c) (d)

Figure 3.2: *Abundance map of Asphalt road (first row) and Dirt (second row): (a) ground-truth, (b) IPLS solution, visual results after running (c) ADMM and (d) PIPA-VM for 11 seconds.*

In order to evaluate the algorithms based on their pointwise convergence, we let them run for a very large number of iterations and compute the relative distance between the current iterate and the solution $\boldsymbol{x}_\infty$. As one can see on Figure 3.3, the sequence generated by PIPA-VM converges faster to the solution than the iterates produced by the other algorithms. It is finally worth noticing that, although an inexact computation of the proximity operator is performed in PIPA-VM, the method appears to be robust to the error generated by this approximation.

§ 3.7 Application: Joint geometry-texture decomposition and reconstruction

3.7.1 Geometry-texture decomposition

Various problems in image processing and computer vision can be formulated as the decomposition of a natural image into texture and geometry components. One can mention, for instance, texture segmentation [Frecon et al., 2016], classification [Aujol and Chan, 2006], or digital inpainting [Bertalmio

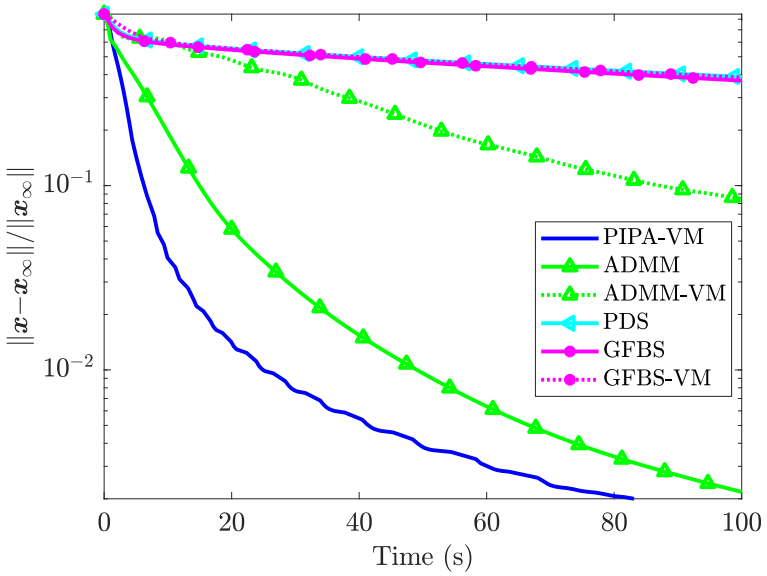


Figure 3.3: *Relative distance from current iterate to limit point as a function of time.*

et al., 2003]. In the following, we will denote by $\boldsymbol{x}^t \in \mathbb{R}^r$ and $\boldsymbol{x}^g \in \mathbb{R}^r$, the texture and geometry components of a natural image $\boldsymbol{x}^{t+g} \in \mathbb{R}^r$, so that $\boldsymbol{x}^{t+g} = \boldsymbol{x}^t + \boldsymbol{x}^g$. The geometry $\boldsymbol{x}^g$ represents a piecewise smooth version of the image, and can be extracted by using the total variation semi-norm [Osher et al., 2003]. The texture highlights local components with higher spatial frequencies. Depending on the considered application, different texture models can be found in the literature, based on wavelet decompositions [Briceño-Arias et al., 2011] or on the Hölder exponent [Pustelnik et al., 2013], to name only a few. Here, we will focus on material images in which the texture is located near the boundaries of different objects, as it can happen for instance in material image analysis, where the samples are subject to erosion and microporosity. Therefore, we will rely on the Laplacian detector for texture extraction, as the latter is known to be useful for edge and blob detection [Haralick, 1979]. Another issue is the acquisition procedure used for this type of images.

3.7.2 X-ray computed tomography

X-Ray Computed Tomography (CT) is a fast non-destructive scanning technique [Kak and Stanley, 2001], which is frequently used to acquire images from material samples. The acquisition process in CT consists in measuring the absorption of an object along a sampled grid of size L for N_θ angular positions. It can be modeled through the discrete Radon projection operator $\boldsymbol{H} \in \mathbb{R}^{n \times r}$, with r the number of pixels and $n = L \times N_\theta$ the number

of measurements. Although matrix $\boldsymbol{H}$ is sparse, it is also high-dimensional and ill-conditioned. Hence, reconstructing the image from the measured data $\boldsymbol{y} \in \mathbb{R}^n$ (also called sinogram) is a challenging inverse problem [Chouzenoux et al., 2013b], which involves ringing artifacts, contrast issues due to beam hardening, and noise caused by sensor motion [Ketcham and Carlson, 2001]. These artifacts can be partially removed by improving scanning techniques or by increasing the number of measurements, but this comes at a price and lots of applications can benefit from a faster subsampled CT acquisition.

Reconstructing the image from the acquired data is a heavy processing step which can introduce a bias in subsequent image processing tasks, such as the classification of material components [Gouillart et al., 2013]. Therefore, we propose to perform jointly two tasks: the reconstruction and the geometry-texture decomposition. We show that this decomposition can be performed in a reasonable time with PIPA.

3.7.3 Problem formulation

We consider the following variational formulation,

$$\begin{array}{ll} \underset{(\boldsymbol{x}^t, \boldsymbol{x}^g) \in \mathbb{R}^r \times \mathbb{R}^r}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{F}\boldsymbol{x}^t\|^2 + \kappa \mathrm{TV}(\boldsymbol{x}^g) \\ \text{subject to} & \boldsymbol{x}^t + \boldsymbol{x}^g \in [x_{\min}, x_{\max}]^r \\ & \boldsymbol{x}^t \in [-\alpha, \alpha]^r \\ & \|\boldsymbol{H}(\boldsymbol{x}^t + \boldsymbol{x}^g) - \boldsymbol{y}\|_\infty \leq \chi \end{array} \tag{3.63}$$

where $x_{\min} = 0$ and $x_{\max} = 1$ are the minimal and maximal pixel intensity values, $\alpha > 0$ is a range value parameter for the texture, $\kappa > 0$ is a regularization parameter, $\chi > 0$ is an upperbound on the measurement uncertainty, TV denotes the isotropic total variation semi-norm with (zero) Dirichlet boundary conditions. Moreover, $\boldsymbol{F} = \boldsymbol{I}_r - \boldsymbol{\Delta} \in \mathbb{R}^{r \times r}$ where $\boldsymbol{\Delta} \in \mathbb{R}^{r \times r}$ is the Laplacian associated with the following 2D kernel padded with circulant assumption:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

The first term in the objective function enforces edge detection in the texture, while the geometry is made piecewise smooth thanks to the total variation regularization. The first set of constraints represents bounds on the pixel values of the natural image. The texture, which is supposed to capture small variations in the image, is modeled as a zero-centered component in the second set of constraints, where we take $\alpha = x_{\max}/3$. The last constraint is the data-fit term, which can be decomposed into $2 \times n$ linear inequalities. Hence, the constraints can be reformulated as $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \in]-\infty, 0]^p$ where

$\boldsymbol{x} = [(\boldsymbol{x}^t)^\top, (\boldsymbol{x}^g)^\top]^\top$, $\boldsymbol{M} \in \mathbb{R}^{p\times m}$, $\boldsymbol{b} \in \mathbb{R}^p$, $p = 2(n+2r)$, $m = 2r$,

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{I}_r \\ -\boldsymbol{I}_r & -\boldsymbol{I}_r \\ \boldsymbol{I}_r & \boldsymbol{0}_{r\times r} \\ -\boldsymbol{I}_r & \boldsymbol{0}_{r\times r} \\ \boldsymbol{H} & \boldsymbol{H} \\ -\boldsymbol{H} & -\boldsymbol{H} \end{pmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{pmatrix} x_{\max}\boldsymbol{1}_r \\ -x_{\min}\boldsymbol{1}_r \\ \alpha\boldsymbol{1}_r \\ \alpha\boldsymbol{1}_r \\ \boldsymbol{y} + \chi\boldsymbol{1}_n \\ -\boldsymbol{y} + \chi\boldsymbol{1}_n \end{pmatrix}. \tag{3.64}$$

As in the previous example, (3.63) is an instance of Problem (3.1) where $(\forall \boldsymbol{x} \in \mathbb{R}^m)\ f(\boldsymbol{x}) = \kappa \mathrm{TV}(\boldsymbol{P}_g \boldsymbol{x})$ and $g(\boldsymbol{x}) = 1/2\|\boldsymbol{F}\boldsymbol{P}_t\boldsymbol{x}\|^2$, with $\boldsymbol{P}_t = (\boldsymbol{I}_r\ \ \boldsymbol{0}_{r\times r})$ and $\boldsymbol{P}_g = (\boldsymbol{0}_{r\times r}\ \ \boldsymbol{I}_r)$. It can be noted that Assumptions 3.2.1 and 3.4.6 are satisfied. Thus, Theorems 3.4.1 and 3.4.5 hold. Since Assumption 3.4.2 does not hold, Theorem 3.4.4 does not apply here. The solution to (3.63) is not necessarily unique. Although the strict complementarity required to apply Theorem 3.4.7 is difficult to check, the convergence of PIPA to a single cluster point was observed in practice.

3.7.4 Initialization

In order to find an initial point that satisfies strictly the constraints, we set $\boldsymbol{x}^t$ to zero. Following the method in [Boyd and Vandenberghe, 2004, Chap. 11.4], we solve the minimization problem below to initialize $\boldsymbol{x}^g$,

$$\begin{array}{ll} \underset{(s,\boldsymbol{x}^g)\in\mathbb{R}\times\mathbb{R}^r}{\text{minimize}} & s \\ \text{subject to} & s \geq 0,\ \boldsymbol{x}^g \in [x_{\min}, x_{\max}]^r \\ & \|\boldsymbol{H}\boldsymbol{x}^g - \boldsymbol{y}\|_\infty \leq \chi + s, \end{array} \tag{3.65}$$

where $s \geq 0$ is the maximal infeasibility. The ground-truth natural image $\overline{\boldsymbol{x}}^{t+g}$ satisfies $\|\boldsymbol{H}\overline{\boldsymbol{x}}^{t+g} - \boldsymbol{y}\|_\infty < \chi$ so that the solution to (3.65) is reached for $s = 0$. Problem (3.65) is a linear programming problem just like problem (2.18). In addition, it is straightforward to check that Assumption 2.2.23 is satisfied. Hence, we can use the Newton barrier method summarized in Algorithm 10 for solving problem (3.65). We use the code from [Boyd and Vandenberghe, 2004, Chap. 11.4] which is available online[2] This algorithm generates iterates that belong to $]x_{\min}, x_{\max}[^r$ so that, in our numerical experiments, we are able to find a strictly feasible initial point in a reasonable time.

3.7.5 Variable metric and hyperparameters

Let $\mu_j \in \mathbb{R}_+^*$ and $k \in \mathbb{N}$. For every $\boldsymbol{x} \in \mathcal{D}$, the Hessian of φ_{μ_j} at $\boldsymbol{x} \in \mathbb{R}^m$ is equal to

$$\nabla^2 \varphi_{\mu_j}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{F}^\top\boldsymbol{F} + \mu_j \mathrm{Diag}(d_3(\boldsymbol{x})) + G(\boldsymbol{x}) & G(\boldsymbol{x}) \\ G(\boldsymbol{x}) & G(\boldsymbol{x}) \end{pmatrix}, \tag{3.66}$$

[2] https://web.stanford.edu/~boyd/cvxbook/cvxbook_examples/chap11/

where

$$G(\boldsymbol{x}) = \mu_j \left(\mathrm{Diag}(d_1(\boldsymbol{x})) + \boldsymbol{H}^\top \mathrm{Diag}(d_2(\boldsymbol{x}))\boldsymbol{H} \right). \tag{3.67}$$

Hereabove, $\mathrm{Diag}(d_1(\boldsymbol{x}))$, $\mathrm{Diag}(d_2(\boldsymbol{x}))$ and $\mathrm{Diag}(d_3(\boldsymbol{x}))$ are the diagonal matrices whose elements are given by $d_1(\boldsymbol{x}) \in \mathbb{R}^r$, $d_2(\boldsymbol{x}) \in \mathbb{R}^n$, and $d_3(\boldsymbol{x}) \in \mathbb{R}^r$, respectively. These vectors are defined as follows.

$$\begin{aligned}
&(\forall i \in \{1,\dots,r\}) \qquad (d_1(\boldsymbol{x}))_i = \left(\left(\boldsymbol{x}^{t+g}\right)_i - x_{\min} \right)^{-2} + \left(x_{\max} - \left(\boldsymbol{x}^{t+g}\right)_i \right)^{-2} \\
&(\forall j \in \{1,\dots,n\}) \qquad (d_2(\boldsymbol{x}))_j = \left(\left(\boldsymbol{H}\boldsymbol{x}^{t+g} - \boldsymbol{y}\right)_j + \chi \right)^{-2} + \\
&\qquad\qquad\qquad\qquad\qquad\qquad \left(\left(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}^{t+g}\right)_j + \chi \right)^{-2} \\
&(\forall i \in \{1,\dots,r\}) \qquad (d_3(\boldsymbol{x}))_i = \left(\left(\boldsymbol{x}^{t}\right)_i + \alpha \right)^{-2} + \left(\alpha - \left(\boldsymbol{x}^{t}\right)_i \right)^{-2}
\end{aligned}$$

Given the huge size and ill-conditioning of $\boldsymbol{H}$, the inversion of $\nabla^2 \varphi_{\mu_j}(\boldsymbol{x})$ is hardly feasible. Hence, instead of using the full Hessian of φ_{μ_j} for the variable metric as in Section 3.6, we propose to use an upper bound of it, i.e. $\boldsymbol{M}_k \in \mathfrak{S}_m^+$ such that $\boldsymbol{M}_k - \nabla^2 \varphi_{\mu_j}(\boldsymbol{x}_k)$ also belongs to $\mathfrak{S}_m^+$. We propose to majorize $\mu_j \mathrm{Diag}(d_3(\boldsymbol{x}))$ by $\beta(\boldsymbol{x})\boldsymbol{I}_r$ where

$$\beta(\boldsymbol{x}) = \max_{1 \le i \le r} \mu_j (d_3(\boldsymbol{x}))_i.$$

For $\boldsymbol{H}^\top \mathrm{Diag}(d_2(\boldsymbol{x}))\boldsymbol{H}$, we propose to follow the strategy in [Chouzenoux et al., 2014c] and upper-bound it by the diagonal matrix $\mathrm{Diag}(\boldsymbol{P}^\top d_2(\boldsymbol{x})) \in \mathbb{R}^{r \times r}$, where $\boldsymbol{P} \in \mathbb{R}^{n \times r}$ is such that for every $i \in \{1,\dots,r\}$ and $j \in \{1,\dots,n\}$,

$$P_{j,i} = H_{j,i} \sum_{s=1}^{r} H_{j,s}.$$

This leads to the following variable metric in Algorithm 11,

$$(\forall k \in \mathbb{N}) \ \boldsymbol{M}_k = \begin{pmatrix} \boldsymbol{F}^\top \boldsymbol{F} + \beta(\boldsymbol{x}_k)\boldsymbol{I}_r + D(\boldsymbol{x}_k) & D(\boldsymbol{x}_k) \\ D(\boldsymbol{x}_k) & D(\boldsymbol{x}_k) \end{pmatrix} \tag{3.68}$$

where

$$D(\boldsymbol{x}_k) = \mu_j \left(\mathrm{Diag}(d_1(\boldsymbol{x}_k)) + \mathrm{Diag}(\boldsymbol{P}^\top d_2(\boldsymbol{x}_k)) \right).$$

Since $D(\boldsymbol{x}_k)$ is diagonal, the operator $\boldsymbol{M}_k$ is straightforward to invert using the Schur formula.

In addition, similarly to Section 3.6, we deduce from Remark 3.5.7 that matrix (3.68) satisfies the boundedness condition required in Algorithm 11. In order to compute the proximity operator of f in such variable metric, we use Algorithm 7. Regarding the hyperparameters in the proposed method, in order to satisfy Assumption 3.3.1, as in the previous example, we take sequences of the form (3.62) with $\mu_0 = 10^{-3}$, $\zeta = 1 + 10^{-5}$, $\bar{\epsilon} = 8.3 \times 10^3$ and $\rho_j = 1.1$ for every $j \in \mathbb{N}$.

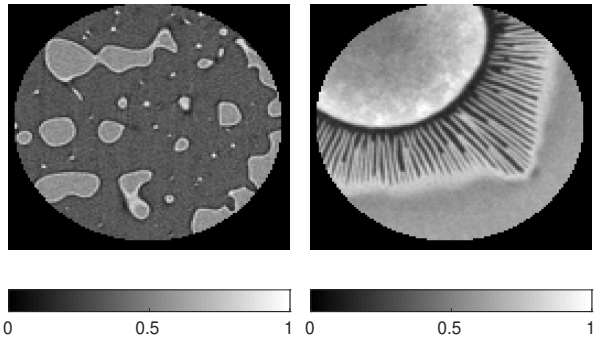


Figure 3.4: *Natural images: (left) phase-separated barium borosilicate glass sample, imaged at the ESRF synchrotron (courtesy of David Bouttes), (right) mushroom Agaricus bisporus (courtesy of DigiMorph.org, The University of Texas High-Resolution X-ray CT Facility (UTCT), and NSF grant IIS-0208675).*

3.7.6 Test settings

We perform the joint reconstruction and decomposition of two high-quality scans, referred to as Glass and Agaricus, which are displayed in Figure 3.4. These images are of size $r = 128 \times 128$. The discrete Radon operator $\boldsymbol{H}$ models parallel projections along $N_\theta = 180$ angular positions on a detector grid of size $L = 128$, so that $n = 180 \times 128$. To account for measurement uncertainty, the sinograms are degraded with a uniform noise with an amplitude χ equal to 2% of the maximal entry of $\boldsymbol{y}$. We set manually the regularization parameter κ so that it leads to a visually satisfatory decomposition: it is set to 0.25 for Glass and to 0.5 for Agaricus. The proposed algorithm PIPA-VM is compared to ADMM summarized in Algorithm 8, which was the most competitive method in Section 3.6. Remark that, in order to make the implementation of ADMM feasible, we follow the same strategy as in [Iordache et al., 2012], and alternate the minimization on the splitting variables. In our example, we need seven splitting variables. Since ADMM does not require a feasible starting point, we run it with two different initializations: ADMM1 refers to ADMM initialized like PIPA-VM, and ADMM2 refers to ADMM initialized with $\boldsymbol{x}^t$ taken as the zero vector and $\boldsymbol{x}^g$ set to $1/2(x_{\min} + x_{\max})\mathbf{1}_r$.

All computational times are given for experiments run on Matlab 2018b on an Intel Xeon CPU E5-1650 at frequency 3.20 GHz.

3.7.7 Results

To compare the convergence speed of the different methods, we plot for each of them the relative distance between the current iterate $\boldsymbol{x}$ and the final solution $\boldsymbol{x}_\infty$, obtained after running the algorithms for 12 hours. As

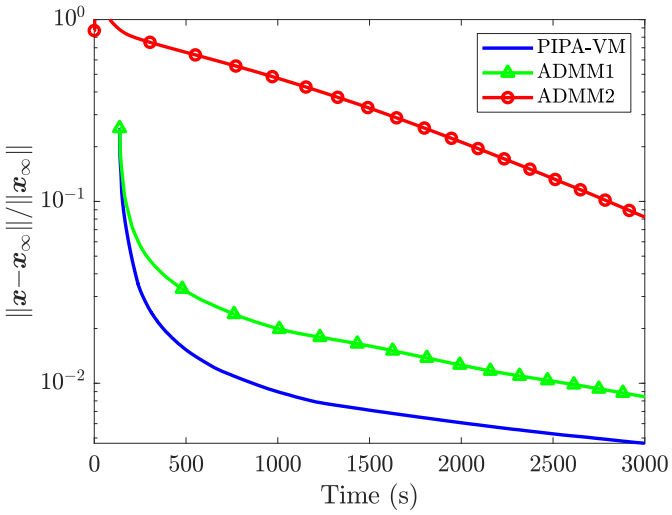


Figure 3.5: *Relative distance from the iterates to the limit point as a function of time for Glass.*

one can see in Figures 3.5 and 3.6, PIPA-VM converges faster to its limit point than ADMM for both initializations. Remark that the time necessary to solve (3.65) and to find a feasible point is taken into account in the graphs. The results clearly show the advantage of using a feasible starting point over a simple initial guess.

To assess the visual quality of the geometry-texture decomposition we consider the solution obtained after reaching the stopping criterion $\|\boldsymbol{x} - \boldsymbol{x}_\infty\|/\|\boldsymbol{x}_\infty\| \leq 10^{-2}$. This accuracy is reached first for PIPA-VM after 14 min for Glass and 18 min for Agaricus. The corresponding visual decomposition and reconstruction after these durations are given in Figure 3.7. As one can see in this figure, PIPA-VM identifies correctly the geometry as an almost piecewise-constant image, free from locally-fast varying components like the gills in the Agaricus mushroom. Moreover, the texture obtained for Glass image captures well the elements on the borders between the two species in presence, and the Agaricus gills can be found in the texture (Figure 3.7 bottom left).

Finally, we evaluate the reconstruction quality based on the signal-to-noise ratio:

$$\mathrm{SNR} = 20\log_{10}(\|\overline{\boldsymbol{x}}^{t+g}\|/\|\boldsymbol{x}^{t+g} - \overline{\boldsymbol{x}}^{t+g}\|), \tag{3.69}$$

where $\overline{\boldsymbol{x}}^{t+g}$ denotes the ground-truth image. The SNR values obtained with the three methods for the reconstructions $\boldsymbol{x}^{t+g}$ after the same durations are summarized in Table 3.2. This table shows that the SNR obtained with PIPA-VM is better than with the other methods after the same durations.

It can be further observed in Figures 3.8 and 3.9, that even if the SNR

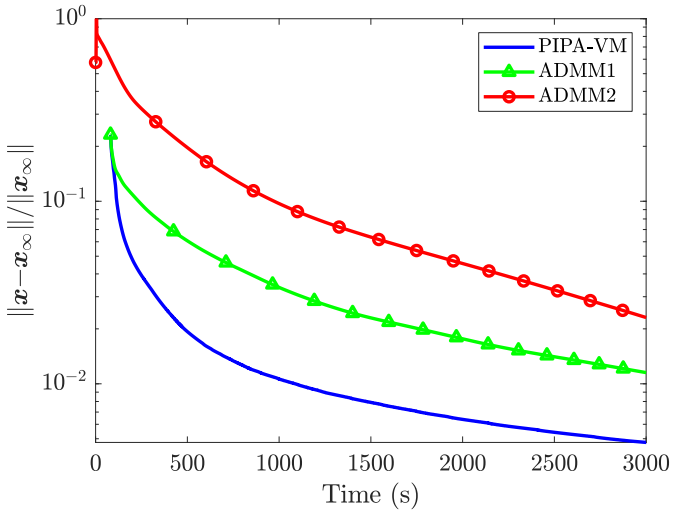


Figure 3.6: *Relative distance from the iterates to the limit point as a function of time for Agaricus.*

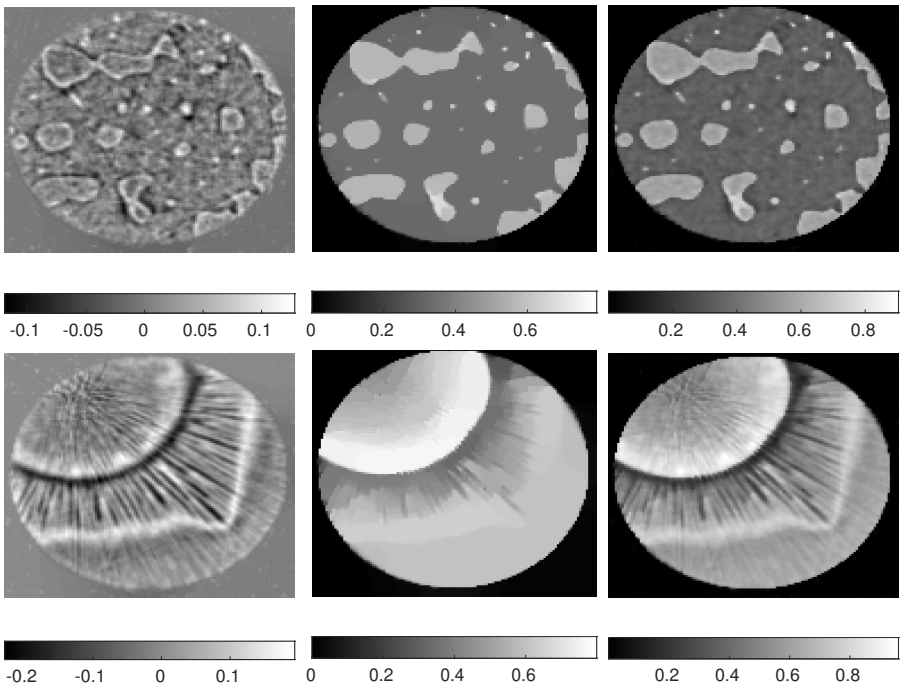


Figure 3.7: *Visual results for PIPA-VM. (Top) Glass obtained after 14 min. (Bottom) Agaricus obtained after 18 min. Left to right: texture, geometry, reconstruction* $\boldsymbol{x}^{t+g}$.

	Glass (14 min)	Agaricus (18 min)
PIPA-VM	**19.0**	**20.57**
ADMM1	18.74	20.32
ADMM2	18.69	20.32

Table 3.2: *SNR (dB) of the reconstruction $\boldsymbol{x}^{t+g}$ obtained after running the algorithms for the same duration.*

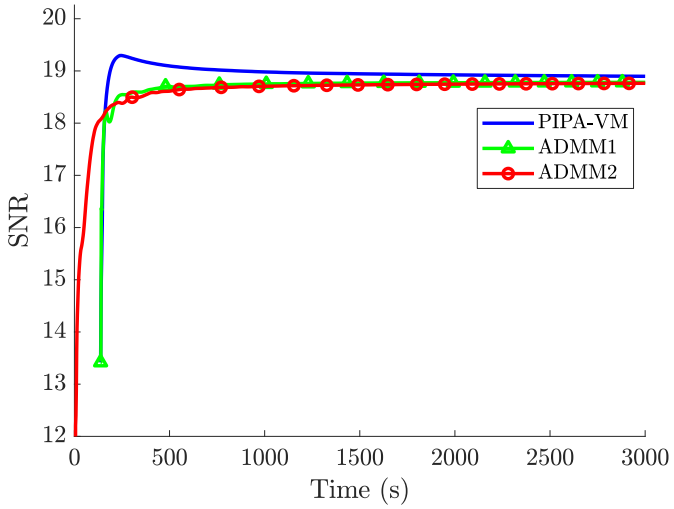


Figure 3.8: *SNR (dB) for $\boldsymbol{x}^{t+g}$ as a function of time for Glass.*

converges to the same value for the three methods, PIPA-VM follows a path which would lead to a better SNR if stopped before convergence.

§ 3.8 Summary

In this chapter, we have shown that it is possible to combine efficiently two powerful optimization frameworks: proximal splitting methods and interior point algorithms. One interesting feature of the resulting iterative method is the use of a variable metric, which can boost the convergence, as illustrated in our hyperspectral unmixing application. The convergence of the proposed method, as well as a convergence rate for the inner loop, have been obtained under suitable assumptions. Regarding numerical performance, as shown on two large-scale image processing applications, our method compares favorably in terms of speed of convergence with state-of-the-art algorithms.

Although we are able to produce fast algorithms for solving optimization problems, they are often sensitive to hyperparameters that need to be set

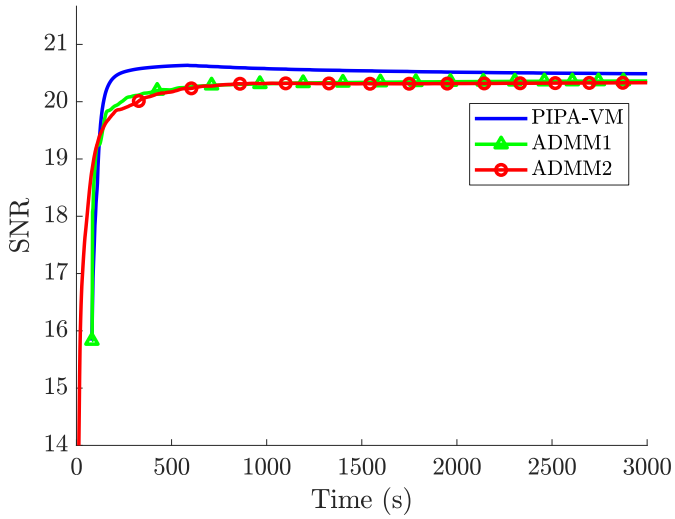


Figure 3.9: *SNR (dB) for $\boldsymbol{x}^{t+g}$ as a function of time for Agaricus.*

carefully. Moreover, variational strategies for addressing inverse problems may suffer from downsides, some of which are mentioned in Section 2.1.2.2. In that regard, neural networks appear as a powerful alternative, which is studied in the next chapter.

- Chapter 4 -

An optimization-inspired neural network architecture for image deblurring

As we mentioned in Section 2.1.4.2, neural networks demonstrate state-of-the-art performance in several applications linked to inverse problems. However, these methods suffer from important downsides including a lack of interpretability, few mathematical results, and no real possibility for imposing constraints on their parameters. On the other hand, variational formulations, whose limitations are discussed in Section 2.1.2.2, allow to incorporate some a priori knowledge or desirable properties for the solution, and the convergence results regarding classical optimization algoritms are well established. Hence, DNNs and variational-based methods can be considered as complementary. One straightforward way to combine the benefits of both approaches is to *unfold* an iterative method and untie the parameters of both the model and the algorithm across the layers of the network [Hershey et al., 2014].

In this chapter, we propose a novel neural network architecture called iRestNet, which is obtained by unfolding a proximal interior point algorithm over a finite number of iterations. More information about deep unfolding methods can be found in Section 2.1.4.3. One key feature of this algorithm is that it produces only feasible iterates thanks to a logarithmic barrier. This barrier enables prior knowledge to be directly incorporated into iRestNet and, as opposed to a projection onto the feasible set, it allows differentiation and gradient backpropagation throughout the network. Hence, gradient descent can be used for training. The stepsize, barrier parameter, and regularization weight are untied across the network and learned for each layer. Thus, once the network has been trained, its application on test images requires only a short execution time per image without any parameter search, as opposed to traditional variational methods.

This chapter is organized as follows. First, we present our contributions with regards to related works in Section 4.1, then, in Section 4.2, we describe

the algorithm which is at the core of our method. Since the proposed algorithm requires the computation of the proximity operator of the barrier, we provide the latter for three useful cases in Section 4.3. The proposed neural network architecture and its associated backpropagation method are presented in Section 4.4. Then, in Section 4.5, we conduct a stability analysis of the proposed network when the data fidelity term and the regularization function are quadratic. Section 4.6 is dedicated to numerical experiments and comparison to state-of-the-art methods for image deblurring; finally, some conclusions are drawn in Section 4.7.

§ 4.1 Link with related works

Several recent works consider replacing handcrafted algorithms by learned iterative methods [Andrychowicz et al., 2016; Li and Malik, 2016]. In these approaches, the goal is to find the minimizer of a given objective function, whereas, in the proposed method, the architecture is inspired by an optimization strategy applied to the minimization of an objective function, but a better indicator of perceptual quality is optimized during the training step.

Only a few works so far have considered combining IPMs with deep learning. Every layer of the network from [Amos and Kolter, 2017] solves a small quadratic problem using an IPM, while in [Trafalis et al., 1997], hard constraints are enforced on weights by using the logarithmic barrier function during training. More recently, an interior point strategy was used to design a recurrent network, whose purpose is to solve a specific convex constrained problem [Krasopoulos and Maratos, 2014]. In our case, however, we have two distinct objective functions. The first one leads to a constrained problem from which the proposed architecture is inspired, while the second one is used during training as a loss function. It is worth noting that the output of the trained network is not necessarily a minimizer of the first objective. Moreover, the second objective could not be a substitute to the first one since it requires the knowledge of the ground-truth, which is available for training time but not in testing conditions. In addition, iRestNet appears to have more flexibility since the regularization weight can vary among layers.

To the best of our knowledge, this chapter presents the first architecture corresponding to a deep unfolded version of an interior point algorithm with untied stepsize and regularization parameter. As opposed to other unfolding methods like [Mardani et al., 2017; Diamond et al., 2017], the proximity operator and the regularization term are kept explicit, which establishes a direct relation between the original algorithm and the network. Other contributions of this work include the expression of the required proximity operator, and of its corresponding gradient, for three standard variational formulations, along with numerical experiments demonstrating the benefit of using the proposed approach over other machine learning and variational

methods for image deblurring.

§ 4.2 Proposed algorithm

4.2.1 Variational formulation and assumptions

Let us focus on inverse problems related to model (2.1), where the observation model is linear. Under this assumption, (2.1) becomes

$$\boldsymbol{y} = \mathcal{D}(\boldsymbol{H}\overline{\boldsymbol{x}}), \tag{4.1}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the observed data, $\overline{\boldsymbol{x}} \in \mathbb{R}^m$ is the sought signal or image, $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ is the observation operator, and $\mathcal{D}$ is the noise perturbation operator. The linear operator $\boldsymbol{H}$ is assumed to be known from a physical model or prior identification step [Lagendijk and Biemond, 2005; Xu and Jia, 2010].

As detailed in Section 2.1.2, the sought image $\overline{\boldsymbol{x}}$ can be classically approximated by the minimizer of a penalized cost function expressed as the sum of a data-fitting term, which measures the fidelity of the solution to the observation model (4.1), and a regularization term, which is introduced so as to avoid meaningless solutions and improve stability relative to noise. This leads to problem (2.3), which is rewritten below in the case of a linear observation model.

$$\underset{\boldsymbol{x}\in\mathcal{C}}{\text{minimize}} \;\; f(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{y}) + \lambda\mathcal{R}(\boldsymbol{x}) \tag{4.2}$$

In the remaining of this chapter, we will assume that, for every $\boldsymbol{y} \in \mathbb{R}^n$, $f(\cdot, \boldsymbol{y}) \in \Gamma_0(\mathbb{R}^n)$ and $\mathcal{R} \in \Gamma_0(\mathbb{R}^m)$ are twice-differentiable functions. Note that such assumption is necessary to define the derivative steps involved in the backpropagation procedure for the training of our network. The feasible set $\mathcal{C}$ is defined by p inequality constraints, which enforce the fulfillment of some properties that are expected to be satisfied a priori by the image:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \dots, p\})\; c_i(\boldsymbol{x}) \leq 0\}, \tag{4.3}$$

where, for every $i \in \{1, \dots, p\}$, $c_i \in \Gamma_0(\mathbb{R}^m)$. The strict interior of the feasible domain is equal to

$$\mathcal{D} = \{\boldsymbol{x} \in \mathbb{R}^m \mid (\forall i \in \{1, \dots, p\})\; c_i(\boldsymbol{x}) < 0\},$$

and it is assumed to be nonempty. The interest of such constraints in the resolution of inverse problems has been discussed in the previous chapter in Remark 3.2.2.

Finally, we will assume that either $f(\boldsymbol{H}\cdot, \boldsymbol{y}) + \lambda\mathcal{R}$ is coercive, or $\mathcal{C}$ is bounded. Then the existence of solutions for (4.2) is guaranteed. It is worthy

to emphasize that a large class of penalized formulations encountered in the literature of image restoration fulfills the above requirements, see e.g. [Durand and Nikolova, 2006] and references therein. For simplicity, for all $(\boldsymbol{x}, \boldsymbol{y}, \lambda) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}_+$, we define

$$h(\boldsymbol{x}, \boldsymbol{y}, \lambda) = f(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{y}) + \lambda \mathcal{R}(\boldsymbol{x}),$$

and

$$\nabla_1 h(\boldsymbol{x}, \boldsymbol{y}, \lambda) = \boldsymbol{H}^\top \nabla_1 f(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{y}) + \lambda \nabla \mathcal{R}(\boldsymbol{x}),$$

where $\nabla_1 f$ is the partial gradient of f with respect to its first variable.

In general, problem (4.2) does not have a closed-form solution on account of the inequality constraints, even for simple regularizations, hence an iterative solver must be used. Several resolution approaches are available, either based on projected gradient strategies [Iusem, 2003; Bonettini and Prato, 2015a], ADMM [Boyd et al., 2011], primal-dual schemes [Komodakis and Pesquet, 2015], or interior point techniques [Bonettini and Serafini, 2009b]. As it is shown in Chapter 3, combining the interior point framework with a proximal forward–backward strategy leads to very competitive solvers for inverse problems. As detailed in Section 2.2.5, the idea behind IPMs is to replace the initial constrained optimization problem by a sequence of unconstrained subproblems of the form:

$$\min_{\boldsymbol{x} \in \mathbb{R}^m} \; f(\boldsymbol{H}\boldsymbol{x}, \boldsymbol{y}) + \lambda \mathcal{R}(\boldsymbol{x}) + \mu \mathcal{B}(\boldsymbol{x}) \tag{4.4}$$

where $\mu \in \mathbb{R}_+^*$ is the barrier parameter and $\mathcal{B} : \mathbb{R}^m \to]-\infty, +\infty]$ is the logarithmic barrier function whose definition is recalled below.

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \mathcal{B}(\boldsymbol{x}) = \begin{cases} -\displaystyle\sum_{i=1}^{p} \ln(-c_i(\boldsymbol{x})) & \text{if } \boldsymbol{x} \in \mathcal{D}, \\ +\infty & \text{otherwise.} \end{cases} \tag{4.5}$$

We have made the assumption that either $f(\boldsymbol{H}\cdot, \boldsymbol{y}) + \lambda\mathcal{R}$ is coercive, or $\mathcal{C}$ is bounded, hence, the set of solutions to (4.2) is bounded. Since $\mathcal{D}$ is not empty, we can apply Corollary 3.5.1, which ensures the existence of solutions to (4.4).

4.2.2 Proposed iterative schemes

Thanks to the proximity operator, the IPM from [Kaplan and Tichatschke, 1998] does not require any matrix inversion, as opposed to the classical Newton barrier method. When the proximity operator is computed in an exact manner, the proposed IPM can be rewritten as Algorithm 13, whose convergence has been proven under some assumptions [Kaplan and Tichatschke, 1998, Theorem 4.1].

Algorithm 13: Exact version of the proximal IPM in [Kaplan and Tichatschke, 1998] applied to problem (4.2).

Let $\underline{\gamma} \in \mathbb{R}_+^*$ and $(\gamma_k)_{k\in\mathbb{N}}$ be a sequence such that $(\forall k \in \mathbb{N})\ \underline{\gamma} \leq \gamma_k$;
Initialization: Let $\boldsymbol{x}_0 \in \mathcal{D}$;
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{x}_{k+1} = \operatorname{prox}_{\gamma_k(h(\cdot,\boldsymbol{y},\lambda)+\mu_k\mathcal{B})}(\boldsymbol{x}_k)$
end

Algorithm 13 requires evaluating the proximity operator of the sum of the barrier and the regularized cost function, which can be an issue since, in most of the cases, this operator does not have a closed-form expression. This is the reason why we propose to modify it by introducing a forward step, which leads to Algorithm 14.

Algorithm 14: Proposed forward–backward proximal IPM.

Let $\underline{\gamma} \in \mathbb{R}_+^*$ and $(\gamma_k)_{k\in\mathbb{N}}$ be a sequence such that $(\forall k \in \mathbb{N})\ \underline{\gamma} \leq \gamma_k$;
Initialization: Let $\boldsymbol{x}_0 \in \mathcal{D}$;
for $k = 0, 1, \ldots$ **do**
  $\boldsymbol{x}_{k+1} = \operatorname{prox}_{\gamma_k\mu_k\mathcal{B}}(\boldsymbol{x}_k - \gamma_k\nabla_1 h(\boldsymbol{x}_k, \boldsymbol{y}, \lambda))$
end

To the best of our knowledge, there is no available convergence study for Algorithm 14 among the literature of interior-point methods. There exist links between the above algorithm and the diagonal or penalization method introduced in [Czarnecki et al., 2016]. Indeed, taking $A \equiv 0$ and $\Psi_1 \equiv 0$ in [Czarnecki et al., 2016] leads to Algorithm 14, whose convergence is proven. However, there are some key differences between both approaches, namely *i)* in [Czarnecki et al., 2016], the barrier parameter tends to infinity while it goes to zero in our case, and *ii)* the algorithm in [Czarnecki et al., 2016] solves a hierarchical minimization problem instead of the constrained optimization problem (4.2). It is worth noting that Algorithm 14 only requires computing the proximity operator of the logarithmic barrier. We will provide its expression in Section 4.3 for three different types of constraints.

4.2.3 Limitations

In IPMs, the barrier parameter and stepsize sequences, $(\mu_k)_{k\in\mathbb{N}}$ and $(\gamma_k)_{k\in\mathbb{N}}$, are usually set by following some heuristic rules, which ensure the convergence of the method to a minimizer of the considered objective function. However, handcrafted variational formulations do not necessarily capture perceptual image quality well. These heuristics can thus lead to a loss in

terms of efficiency and versatility of the resulting restoration schemes. Moreover, as already mentioned, an accurate setting of the regularization weights is particularly critical in order to obtain a satisfactory image quality when using such penalized restoration approaches. Existing approaches for selecting λ, which are based on statistical considerations, are usually associated with a substantial increase of the computational cost.

To overcome these limitations, we propose to unfold Algorithm 14 over a given number of iterations and to learn the stepsize, the barrier and the regularization parameters for every iteration in a supervised fashion. Our machine learning method will make use of gradient backpropagation for its training step. The latter requires the derivatives of the proximity operator in Algorithm 14 with respect to its input and to the aforementioned parameters which are to be learned. Therefore, we first conduct an analysis of the proximity operator of the barrier and of its derivatives, for three examples of interest in Section 4.3.

§ 4.3 Proximity operator of the barrier

Let $\mathcal{B}$ be defined as in (4.5) and for all $\mu \in \mathbb{R}_+^*$, $\gamma \in \mathbb{R}_+^*$ and $\boldsymbol{x} \in \mathbb{R}^m$, let φ be defined as follows:

$$\varphi(\boldsymbol{x}, \mu, \gamma) = \operatorname{prox}_{\gamma\mu\mathcal{B}}(\boldsymbol{x}).$$

We provide in this section expressions of φ and of its derivatives with respect to its input variable $\boldsymbol{x}$ and the involved barrier and stepsize parameters (μ, γ), for three common types of constraints. The latter will be necessary for training the proposed neural network using a gradient backpropagation scheme.

4.3.1 Affine constraints

Let us first consider the following half-space constraint:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^m \mid \boldsymbol{a}^\top \boldsymbol{x} \leq b\}, \tag{4.6}$$

with $\boldsymbol{a} \in \mathbb{R}^m \setminus \{\boldsymbol{0}_m\}$ and $b \in \mathbb{R}$.

Proposition 4.3.1 *Let $\gamma \in \mathbb{R}_+^*$, $\mu \in \mathbb{R}_+^*$, and let $\mathcal{B}$ be the function associated to (4.6), defined as*

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \mathcal{B}(\boldsymbol{x}) = \begin{cases} -\ln(b - \boldsymbol{a}^\top \boldsymbol{x}) & \text{if } \boldsymbol{a}^\top \boldsymbol{x} < b, \\ +\infty & \text{otherwise.} \end{cases}$$

Then, for every $\boldsymbol{x} \in \mathbb{R}^m$, the proximity operator of $\gamma\mu\mathcal{B}$ at $\boldsymbol{x}$ is given by

$$\varphi(\boldsymbol{x}, \mu, \gamma) = \boldsymbol{x} + \frac{b - \boldsymbol{a}^\top \boldsymbol{x} - \sqrt{(b - \boldsymbol{a}^\top \boldsymbol{x})^2 + 4\gamma\mu\|\boldsymbol{a}\|^2}}{2\|\boldsymbol{a}\|^2} \boldsymbol{a}. \tag{4.7}$$

In addition, the Jacobian matrix of φ with respect to $\boldsymbol{x}$ and the gradients of φ with respect to μ and γ are given by

$$J_{\varphi}^{(\boldsymbol{x})}(\boldsymbol{x},\mu,\gamma) = \boldsymbol{I}_m - \frac{1}{2\|\boldsymbol{a}\|^2}\left(1 + \frac{\boldsymbol{a}^\top\boldsymbol{x} - b}{\sqrt{(b-\boldsymbol{a}^\top\boldsymbol{x})^2 + 4\gamma\mu\|\boldsymbol{a}\|^2}}\right)\boldsymbol{a}\boldsymbol{a}^\top, \quad (4.8)$$

$$\nabla_{\varphi}^{(\mu)}(\boldsymbol{x},\mu,\gamma) = \frac{-\gamma}{\sqrt{(b-\boldsymbol{a}^\top\boldsymbol{x})^2 + 4\gamma\mu\|\boldsymbol{a}\|^2}}\boldsymbol{a}, \quad (4.9)$$

and

$$\nabla_{\varphi}^{(\gamma)}(\boldsymbol{x},\mu,\gamma) = \frac{-\mu}{\sqrt{(b-\boldsymbol{a}^\top\boldsymbol{x})^2 + 4\gamma\mu\|\boldsymbol{a}\|^2}}\boldsymbol{a}. \quad (4.10)$$

Proof. The expression for the proximity operator (4.7) directly follows from Example 2.2.15(i), and Propositions 2.2.12 and 2.2.13. Taking the derivative of (4.7) with respect to $\boldsymbol{x}$, μ and γ leads to (4.8)–(4.10).
□

4.3.2 Hyperslab constraints

We now consider the following hyperslab set:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^m \mid \underline{b} \leq \boldsymbol{a}^\top\boldsymbol{x} \leq \overline{b}\}, \quad (4.11)$$

where $\boldsymbol{a} \in \mathbb{R}^m \setminus \{\boldsymbol{0}_m\}$, $\underline{b} \in \mathbb{R}$ and $\overline{b} \in \mathbb{R}$ with $\underline{b} < \overline{b}$.

Proposition 4.3.2 *Let $\gamma \in \mathbb{R}_+^*$, $\mu \in \mathbb{R}_+^*$, and let $\mathcal{B}$ be the barrier function associated to (4.11), defined as*

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \mathcal{B}(\boldsymbol{x}) = \begin{cases} -\ln(\overline{b} - \boldsymbol{a}^\top\boldsymbol{x}) - \ln(\boldsymbol{a}^\top\boldsymbol{x} - \underline{b}) & \text{if } \underline{b} < \boldsymbol{a}^\top\boldsymbol{x} < \overline{b}, \\ +\infty & \text{otherwise.} \end{cases}$$

Then, for every $\boldsymbol{x} \in \mathbb{R}^m$, the proximity operator of $\gamma\mu\mathcal{B}$ at $\boldsymbol{x}$ is given by

$$\varphi(\boldsymbol{x},\mu,\gamma) = \boldsymbol{x} + \frac{\kappa(\boldsymbol{x},\mu,\gamma) - \boldsymbol{a}^\top\boldsymbol{x}}{\|\boldsymbol{a}\|^2}\boldsymbol{a}, \quad (4.12)$$

where $\kappa(\boldsymbol{x},\mu,\gamma)$ is the unique solution in $]\underline{b},\overline{b}[$, of the following cubic equation:

$$0 = z^3 - (\underline{b} + \overline{b} + \boldsymbol{a}^\top\boldsymbol{x})z^2 + (\underline{b}\overline{b} + \boldsymbol{a}^\top\boldsymbol{x}(\underline{b} + \overline{b}) - 2\gamma\mu\|\boldsymbol{a}\|^2)z - \underline{b}\overline{b}\boldsymbol{a}^\top\boldsymbol{x} + \gamma\mu(\underline{b} + \overline{b})\|\boldsymbol{a}\|^2. \quad (4.13)$$

In addition, the Jacobian matrix of φ with respect to $\boldsymbol{x}$ and the gradients of φ with respect to μ and γ are given by

$$J_{\varphi}^{(\boldsymbol{x})}(\boldsymbol{x},\mu,\gamma) = \boldsymbol{I}_m + \frac{1}{\|\boldsymbol{a}\|^2}\left(\frac{(\overline{b}-\kappa(\boldsymbol{x},\mu,\gamma))(\underline{b}-\kappa(\boldsymbol{x},\mu,\gamma))}{\eta(\boldsymbol{x},\mu,\gamma)} - 1\right)\boldsymbol{a}\boldsymbol{a}^\top, \tag{4.14}$$

$$\nabla_{\varphi}^{(\mu)}(\boldsymbol{x},\mu,\gamma) = \frac{-\gamma(\underline{b}+\overline{b}-2\kappa(\boldsymbol{x},\mu,\gamma))}{\eta(\boldsymbol{x},\mu,\gamma)}\boldsymbol{a}, \tag{4.15}$$

and

$$\nabla_{\varphi}^{(\gamma)}(\boldsymbol{x},\mu,\gamma) = \frac{-\mu(\underline{b}+\overline{b}-2\kappa(\boldsymbol{x},\mu,\gamma))}{\eta(\boldsymbol{x},\mu,\gamma)}\boldsymbol{a}, \tag{4.16}$$

where

$$\begin{aligned}\eta(\boldsymbol{x},\mu,\gamma) = {}& (\overline{b}-\kappa(\boldsymbol{x},\mu,\gamma))(\underline{b}-\kappa(\boldsymbol{x},\mu,\gamma)) \\ & - (\underline{b}+\overline{b}-2\kappa(\boldsymbol{x},\mu,\gamma))(\kappa(\boldsymbol{x},\mu,\gamma)-\boldsymbol{a}^\top\boldsymbol{x}) - 2\gamma\mu\|\boldsymbol{a}\|^2.\end{aligned}$$

Proof. Let $\boldsymbol{x}\in\mathbb{R}^m$, $\gamma\in\mathbb{R}_+^*$, and $\mu\in\mathbb{R}_+^*$. The expression for the proximity operator (4.12) follows from Example 2.2.15(ii) and Proposition 2.2.13. Let F be defined as follows:

$$F(\boldsymbol{x},\mu,\gamma,z) = (\overline{b}-z)(\underline{b}-z)(z-\boldsymbol{a}^\top\boldsymbol{x}) + \gamma\mu(\overline{b}+\underline{b}-2z)\|\boldsymbol{a}\|^2, \tag{4.17}$$

for $z\in]\underline{b},\overline{b}[$. Expanding (4.17) gives the following:

$$\begin{aligned}F(\boldsymbol{x},\mu,\gamma,z) = {}& z^3 - (\boldsymbol{a}^\top\boldsymbol{x}+\underline{b}+\overline{b})z^2 + (\underline{b}\overline{b}+\boldsymbol{a}^\top\boldsymbol{x}(\underline{b}+\overline{b}) - 2\gamma\mu\|\boldsymbol{a}\|^2)z \\ & - \underline{b}\overline{b}\boldsymbol{a}^\top\boldsymbol{x} + \gamma\mu(\underline{b}+\overline{b})\|\boldsymbol{a}\|^2.\end{aligned}$$

Hence, by definition of $\kappa(\boldsymbol{x},\mu,\gamma)$, we have $F(\boldsymbol{x},\mu,\gamma,\kappa(\boldsymbol{x},\mu,\gamma)) = 0$. In addition, the derivative of F with respect to its last variable is equal to

$$\nabla F^{(z)}(\boldsymbol{x},\mu,\gamma,z) = (\overline{b}-z)(\underline{b}-z) - (\underline{b}+\overline{b}-2z)(z-\boldsymbol{a}^\top\boldsymbol{x}) - 2\gamma\mu\|\boldsymbol{a}\|^2.$$

By construction, $(\overline{b}-\kappa(\boldsymbol{x},\mu,\gamma))(\underline{b}-\kappa(\boldsymbol{x},\mu,\gamma)) < 0$. Moreover, $-2\gamma\mu\|\boldsymbol{a}\|^2 < 0$ and, since $F(\boldsymbol{x},\mu,\gamma,\kappa(\boldsymbol{x},\mu,\gamma)) = 0$, it follows that $(\underline{b}+\overline{b}-2\kappa(\boldsymbol{x},\mu,\gamma))$ and $\kappa(\boldsymbol{x},\mu,\gamma)-\boldsymbol{a}^\top\boldsymbol{x}$ share the same sign. Hence,

$$\eta(\boldsymbol{x},\mu,\gamma) = \nabla F^{(z)}(\boldsymbol{x},\mu,\gamma,\kappa(\boldsymbol{x},\mu,\gamma)) \neq 0.$$

From the implicit function theorem [Dontchev and Rockafellar, 2009, Theorem 1B.1], we deduce that the gradient of κ with respect to $\boldsymbol{x}$ and the partial derivatives of κ with respect to μ and γ exist and are equal to

$$\nabla\kappa^{(\boldsymbol{x})}(\boldsymbol{x},\mu,\gamma) = \frac{(\overline{b}-\kappa(\boldsymbol{x},\mu,\gamma))(\underline{b}-\kappa(\boldsymbol{x},\mu,\gamma))}{\eta(\boldsymbol{x},\mu,\gamma)}\boldsymbol{a}, \tag{4.18}$$

$$\nabla\kappa^{(\mu)}(\boldsymbol{x},\mu,\gamma) = \frac{-\gamma\|\boldsymbol{a}\|^2(\underline{b}+\overline{b}-2\kappa(\boldsymbol{x},\mu,\gamma))}{\eta(\boldsymbol{x},\mu,\gamma)}, \tag{4.19}$$

and

$$\nabla\kappa^{(\gamma)}(\boldsymbol{x},\mu,\gamma) = \frac{-\mu\|\boldsymbol{a}\|^2(\underline{b}+\overline{b}-2\kappa(\boldsymbol{x},\mu,\gamma))}{\eta(\boldsymbol{x},\mu,\gamma)}. \tag{4.20}$$

Differentiating (4.12) with respect to $\boldsymbol{x}$, μ and γ and using (4.18)–(4.20) yields (4.14)–(4.16).
□

It can be noted that Example 2.2.15(ii) is a special case of Proposition 4.3.2. The three roots of (4.13) can easily be computed using the Cardano formula.

4.3.3 Bounded ℓ_2-norm

We now consider the case when the feasible set in (4.2) is a Euclidean ball,

$$\mathcal{C} = \{\boldsymbol{x}\in\mathbb{R}^m \mid \|\boldsymbol{x}-\boldsymbol{c}\|^2 \leq \alpha\}, \tag{4.21}$$

with $\alpha\in\mathbb{R}_+^*$ and $\boldsymbol{c}\in\mathbb{R}^m$.

Proposition 4.3.3 *Let $\gamma\in\mathbb{R}_+^*$ and let $\mu\in\mathbb{R}_+^*$. Let $\mathcal{B}$ be the barrier function associated to (4.21), defined as*

$$(\forall\boldsymbol{x}\in\mathbb{R}^m)\quad \mathcal{B}(\boldsymbol{x}) = \begin{cases} -\ln(\alpha-\|\boldsymbol{x}-\boldsymbol{c}\|^2) & \text{if } \|\boldsymbol{x}-\boldsymbol{c}\|^2<\alpha,\\ +\infty & \text{otherwise.}\end{cases}$$

Then, for every $\boldsymbol{x}\in\mathbb{R}^m$, the proximity operator of $\gamma\mu\mathcal{B}$ at $\boldsymbol{x}$ is given by

$$\varphi(\boldsymbol{x},\mu,\gamma) = \boldsymbol{c} + \frac{\alpha-\kappa(\boldsymbol{x},\mu,\gamma)^2}{\alpha-\kappa(\boldsymbol{x},\mu,\gamma)^2+2\gamma\mu}(\boldsymbol{x}-\boldsymbol{c}), \tag{4.22}$$

where $\kappa(\boldsymbol{x},\mu,\gamma)$ is the unique solution in $[0,\sqrt{\alpha}[$ of the cubic equation:

$$0 = z^3 - \|\boldsymbol{x}-\boldsymbol{c}\|z^2 - (\alpha+2\gamma\mu)z + \alpha\|\boldsymbol{x}-\boldsymbol{c}\|. \tag{4.23}$$

In addition, the Jacobian matrix of φ with respect to $\boldsymbol{x}$ and the gradients of φ with respect to μ and γ are given by

$$J_\varphi^{(\boldsymbol{x})}(\boldsymbol{x},\mu,\gamma) = \frac{\alpha-\|\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}\|^2}{\alpha-\|\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}\|^2+2\gamma\mu}M(\boldsymbol{x},\mu,\gamma), \tag{4.24}$$

$$\nabla_\varphi^{(\mu)}(\boldsymbol{x},\mu,\gamma) = \frac{-2\gamma}{\alpha-\|\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}\|^2+2\gamma\mu}M(\boldsymbol{x},\mu,\gamma)(\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}), \tag{4.25}$$

and

$$\nabla_{\varphi}^{(\gamma)}(\boldsymbol{x},\mu,\gamma)=\frac{-2\mu}{\alpha-\|\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}\|^2+2\gamma\mu}M(\boldsymbol{x},\mu,\gamma)(\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}), \tag{4.26}$$

where

$$M(\boldsymbol{x},\mu,\gamma)=\boldsymbol{I}_m-\frac{2(\boldsymbol{x}-\varphi(\boldsymbol{x},\mu,\gamma))(\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c})^\top}{\alpha-3\|\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c}\|^2+2\gamma\mu+2(\varphi(\boldsymbol{x},\mu,\gamma)-\boldsymbol{c})^\top(\boldsymbol{x}-\boldsymbol{c})}. \tag{4.27}$$

Proof. Let $\boldsymbol{x}\in\mathbb{R}^m$, $\gamma\in\mathbb{R}_+^*$, $\mu\in\mathbb{R}_+^*$. Let us first consider the case when $\boldsymbol{c}=0$. We denote with φ_0 the following proximity operator:

$$\varphi_0(\boldsymbol{x},\mu,\gamma)=\operatorname*{argmin}_{\boldsymbol{u}\in\mathcal{D}}\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|^2-\gamma\mu\ln(\alpha-\|\boldsymbol{u}\|^2).$$

Hence, $\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2<\alpha$ and $\varphi_0(\boldsymbol{x},\mu,\gamma)$ is a solution to the following equation:

$$0=\varphi_0(\boldsymbol{x},\mu,\gamma)-\boldsymbol{x}+\frac{2\gamma\mu}{\alpha-\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2}\varphi_0(\boldsymbol{x},\mu,\gamma). \tag{4.28}$$

Since $\alpha-\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2+2\gamma\mu>0$, (4.28) becomes

$$\varphi_0(\boldsymbol{x},\mu,\gamma)=\frac{\alpha-\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2}{\alpha-\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2+2\gamma\mu}\boldsymbol{x}. \tag{4.29}$$

By taking the norm in both sides of (4.29), we deduce that $\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|=\kappa(\boldsymbol{x},\mu,\gamma)$ is a solution to the cubic equation (4.23). Since the proximity operator at a given $\boldsymbol{x}$ is uniquely defined, there exists only one real solution to (4.23) which belongs to $[0,\sqrt{\alpha}[$. Plugging the latter into (4.29) leads to (4.22). The analysis when $\boldsymbol{c}\neq 0$ is deduced from the case $\boldsymbol{c}=0$ by using Proposition 2.2.12: the proximity operator of $\gamma\mu\mathcal{B}$ at $\boldsymbol{x}$ is given by

$$\varphi(\boldsymbol{x},\mu,\gamma)=\boldsymbol{c}+\varphi_0(\boldsymbol{x}-\boldsymbol{c},\mu,\gamma). \tag{4.30}$$

Now we can study the derivatives of φ_0. For every $\boldsymbol{v}\in\mathbb{R}^m$, let F be defined as

$$F(\boldsymbol{x},\mu,\gamma,\boldsymbol{v})=(\alpha-\|\boldsymbol{v}\|^2)(\boldsymbol{v}-\boldsymbol{x})+2\gamma\mu\boldsymbol{v}.$$

The Jacobian of F with respect to its last variable is equal to

$$J_F^{(\boldsymbol{v})}(\boldsymbol{x},\mu,\gamma,\boldsymbol{v})=(\alpha-\|\boldsymbol{v}\|^2+2\gamma\mu)\boldsymbol{I}_m+2(\boldsymbol{x}-\boldsymbol{v})\boldsymbol{v}^\top.$$

Since $\alpha-\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2>0$, according to the Sherman–Morrison Lemma [Bartlett, 1951], $J_F^{(\boldsymbol{v})}(\boldsymbol{x},\mu,\gamma,\varphi_0(\boldsymbol{x},\mu,\gamma))$ is invertible if and only if

$$\alpha-\|\varphi_0(\boldsymbol{x},\mu,\gamma)\|^2+2\gamma\mu+2\varphi_0(\boldsymbol{x},\mu,\gamma)^\top(\boldsymbol{x}-\varphi_0(\boldsymbol{x},\mu,\gamma))\neq 0.$$

Furthermore, it follows from (4.28) that

$$F(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma)) = \mathbf{0}_m. \tag{4.31}$$

Applying $\varphi_0(\boldsymbol{x}, \mu, \gamma)^\top$ on (4.31) leads to $\varphi_0(\boldsymbol{x}, \mu, \gamma)^\top(\boldsymbol{x} - \varphi_0(\boldsymbol{x}, \mu, \gamma)) \geq 0$. In addition, $\alpha - \|\varphi_0(\boldsymbol{x}, \mu, \gamma)\|^2 + 2\gamma\mu > 0$. Hence, $J_F^{(\boldsymbol{v})}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma))$ is invertible and its inverse is given by the Sherman–Morrison formula:

$$J_F^{(\boldsymbol{v})}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma))^{-1} = \frac{1}{\alpha - \|\varphi_0(\boldsymbol{x}, \mu, \gamma)\|^2 + 2\gamma\mu} \times \left[\boldsymbol{I}_m - \frac{2(\boldsymbol{x} - \varphi_0(\boldsymbol{x}, \mu, \gamma))\varphi_0(\boldsymbol{x}, \mu, \gamma)^\top}{\alpha - 3\|\varphi_0(\boldsymbol{x}, \mu, \gamma)\|^2 + 2\gamma\mu + 2\varphi_0(\boldsymbol{x}, \mu, \gamma)^\top \boldsymbol{x}}\right].$$

From the implicit function theorem [Dontchev and Rockafellar, 2009, Theorem 1B.1] we deduce that the Jacobian of φ_0 with respect to $\boldsymbol{x}$ and the gradients of φ_0 with respect to μ and γ exist and are equal to

$$J_{\varphi_0}^{(\boldsymbol{x})}(\boldsymbol{x}, \mu, \gamma) = -J_F^{(\boldsymbol{v})}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma))^{-1} J_F^{(\boldsymbol{x})}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma)),$$

$$\nabla_{\varphi_0}^{(\mu)}(\boldsymbol{x}, \mu, \gamma) = -J_F^{(\boldsymbol{v})}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma))^{-1} \nabla_F^{(\mu)}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma)),$$

and

$$\nabla_{\varphi_0}^{(\gamma)}(\boldsymbol{x}, \mu, \gamma) = -J_F^{(\boldsymbol{v})}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma))^{-1} \nabla_F^{(\gamma)}(\boldsymbol{x}, \mu, \gamma, \varphi_0(\boldsymbol{x}, \mu, \gamma)).$$

When $c \neq 0$, the derivatives of φ are deduced from those of φ_0 using (4.30):

$$J_{\varphi}^{(\boldsymbol{x})}(\boldsymbol{x}, \mu, \gamma) = -J_F^{(\boldsymbol{v})}(\boldsymbol{x} - \boldsymbol{c}, \mu, \gamma, \varphi(\boldsymbol{x}, \mu, \gamma) - \boldsymbol{c})^{-1} \times J_F^{(\boldsymbol{x})}(\boldsymbol{x} - \boldsymbol{c}, \mu, \gamma, \varphi(\boldsymbol{x}, \mu, \gamma) - \boldsymbol{c}),$$

$$\nabla_{\varphi}^{(\mu)}(\boldsymbol{x}, \mu, \gamma) = -J_F^{(\boldsymbol{v})}(\boldsymbol{x} - \boldsymbol{c}, \mu, \gamma, \varphi(\boldsymbol{x}, \mu, \gamma) - \boldsymbol{c})^{-1} \times \nabla_F^{(\mu)}(\boldsymbol{x} - \boldsymbol{c}, \mu, \gamma, \varphi(\boldsymbol{x}, \mu, \gamma) - \boldsymbol{c}),$$

and

$$\nabla_{\varphi}^{(\gamma)}(\boldsymbol{x}, \mu, \gamma) = -J_F^{(\boldsymbol{v})}(\boldsymbol{x} - \boldsymbol{c}, \mu, \gamma, \varphi(\boldsymbol{x}, \mu, \gamma) - \boldsymbol{c})^{-1} \times \nabla_F^{(\gamma)}(\boldsymbol{x} - \boldsymbol{c}, \mu, \gamma, \varphi(\boldsymbol{x}, \mu, \gamma) - \boldsymbol{c}),$$

which lead to (4.24)-(4.26).
□

Similarly to the previous case, the three solutions to (4.23) can be obtained thanks to the Cardano formula. The form of the resulting proximity

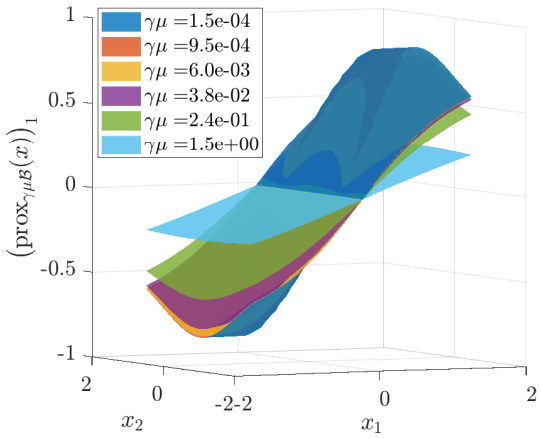


Figure 4.1: *Proximity operator of the logarithmic barrier:* $\left(\text{prox}_{\gamma\mu\mathcal{B}}(\boldsymbol{x})\right)_1$ *for a constraint on the* ℓ_2*-norm as in Section 4.3.3 with* $\alpha = 0.7$.

operator for $m = 2$ is plotted on Figure 4.1 for $\alpha = 0.7$, $\boldsymbol{c} = 0$, and several values of $\gamma\mu$ and $\boldsymbol{x}$; for symmetry reasons, only the first component $\left(\text{prox}_{\gamma\mu\mathcal{B}}(\boldsymbol{x})\right)_1$ is represented.

As shown in this section, the proximity operator of the barrier is easily computable and differentiable for several classic types of constraints. Next, we detail the proposed approach in Section 4.4.

§ 4.4 iRestNet architecture

4.4.1 Overview

Our proposal is to adopt a supervised learning strategy in order to determine, from a training set of images, an optimal setting for the parameters of Algorithm 14, which should lead to an optimal image restoration quality. To this aim, Algorithm 14 is unfolded over K iterations and the regularization parameter λ is untied across the network, so as to provide more flexibility to the approach [Hershey et al., 2014]. The update rule at a given iteration $k \in \{0, \ldots, K-1\}$ reads

$$\boldsymbol{x}_{k+1} = \mathcal{A}\left(\boldsymbol{x}_k, \mu_k, \gamma_k, \lambda_k\right)$$

with

$$\mathcal{A}\left(\boldsymbol{x}_k, \mu_k, \gamma_k, \lambda_k\right) = \text{prox}_{\gamma_k\mu_k\mathcal{B}}\left(\boldsymbol{x}_k - \gamma_k \nabla_1 h\left(\boldsymbol{x}_k, \boldsymbol{y}, \lambda_k\right)\right). \tag{4.32}$$

For every $k \in \{0, \ldots, K-1\}$, we build the kth *layer* $\mathcal{L}_k$ as the association of three hidden structures, $\mathcal{L}_k^{(\mu)}$, $\mathcal{L}_k^{(\gamma)}$ and $\mathcal{L}_k^{(\lambda)}$, followed by the update $\mathcal{A}$.

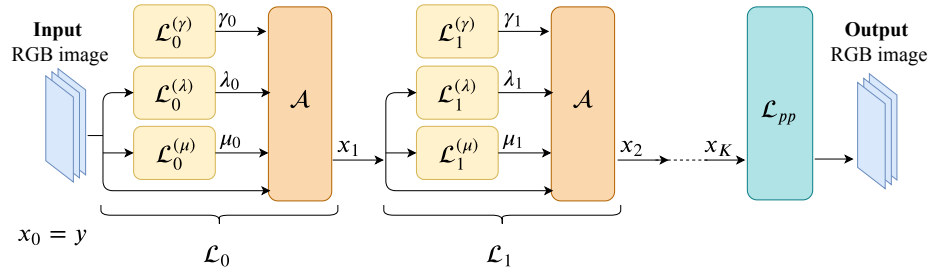


Figure 4.2: *iRestNet global architecture.*

Structures $\mathcal{L}_k^{(\mu)}$, $\mathcal{L}_k^{(\gamma)}$, and $\mathcal{L}_k^{(\lambda)}$ aim at inferring the barrier parameter μ_k, the stepsize γ_k and the regularization weight λ_k, respectively. Since a finite number K of layers (i.e., updates) is used, the convergence of the resulting scheme is not an issue. Note that we also allow in our framework the use of a post–processing step after going through the K layers, that will be denoted as $\mathcal{L}_{\text{pp}}$. The resulting architecture is depicted in Figure 4.2.

4.4.2 Hidden structures

Let us now provide more details about the hidden structures. For every $k \in \{0, \ldots, K-1\}$, the outputs $(\mu_k, \gamma_k, \lambda_k)$ of the structures $\mathcal{L}_k^{(\mu)}$, $\mathcal{L}_k^{(\gamma)}$, and $\mathcal{L}_k^{(\lambda)}$ must be positive. To enforce such constraint, we use the Softplus function [Dugas et al., 2001], defined below, which can be viewed as a smooth approximation of the ReLU activation function:

$$(\forall z \in \mathbb{R}) \quad \text{Softplus}(z) = \ln(1 + \exp(z)).$$

Unlike the ReLU, the gradient of Softplus is never strictly equal to zero, which, given our architecture, helps propagate the gradient through the network. The stepsize is estimated as follows,

$$(\forall k \in \{0, \ldots, K-1\}) \quad \gamma_k = \mathcal{L}_k^{(\gamma)} = \text{Softplus}\left(a_k\right), \tag{4.33}$$

where $(a_k)_{0 \leq k \leq K-1}$ are scalar parameters of the network learned during training. The barrier parameter is obtained using two convolutional and average pooling layers followed by a fully connected layer. The detailed architecture of $\mathcal{L}_k^{(\mu)}$ is depicted in Figure 4.3.

Traditional methods for estimating the regularization parameter generally depend on the signal-to-noise ratio and on the image statistics [Vogel, 2002]. For most applications the noise level is unknown and can be estimated, for instance, by applying a median filter over the wavelet diagonal coefficients of the image [Mallat, 1999; Ramadhan et al., 2017]. This strategy is used in the numerical experiments presented in Section 4.6. The advantage is to yield a network which can handle datasets for which the signal-to-noise

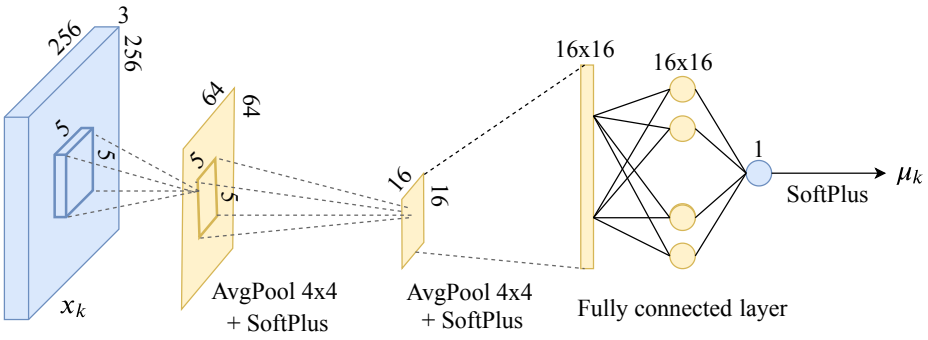


Figure 4.3: *Architecture of* $\mathcal{L}_k^{(\mu)}$.

ratio is unknown and can vary within a reasonable range. The expression of $\mathcal{L}_k^{(\lambda)}$ is then problem–dependent since its expression depends on the regularization function $\mathcal{R}$. A specific example is given in Section 4.6 for the total variation regularization function.

Regarding the post-processing step $\mathcal{L}_{\text{pp}}$, its detailed architecture also depends on the task to be performed. An example is provided in Section 4.6 for the case of deblurring: the purpose of $\mathcal{L}_{\text{pp}}$ is then to remove remaining artifacts using convolutional layers, residual learning, batch normalization, and dilation (the aforementioned notions are defined in Section 2.1.4.1).

4.4.3 Differential calculus

To train the neural network presented in Figure 4.2 using gradient descent, one needs to compute the gradient of $\boldsymbol{x}_K$ with respect to the different parameters of the network. The chain rule can be applied since most of the steps in the network correspond to operators having straightforward derivatives. However, particular care should be taken when differentiating $\mathcal{A}$. Since f and $\mathcal{R}$ are assumed to be twice differentiable, the only area of concern is related to $\text{prox}_{\gamma\mu\mathcal{B}}$. If $\text{prox}_{\gamma\mu\mathcal{B}}$ is simple enough, automatic differentiation [Paszke et al., 2017] can be used. Otherwise, as shown in Section 4.3, for common examples of barrier functions, the differential of this term is well-defined. The corresponding expressions for the derivatives are provided in Propositions 4.3.1–4.3.3.

§ 4.5 Network stability

There is a growing interest for interpretable and stable deep learning architectures [Zhang et al., 2018; Malgouyres and Landsberg, 2019]. One critical issue concerning neural networks is to guarantee that their performance re-

mains acceptable when the input is perturbed. For example, the authors of [Szegedy et al., 2013] show that the class prediction made by AlexNet can be arbitrarily changed by using small nonrandom perturbations on the test image. For some applications involving high risk and legal responsibility, for instance in medical image processing, the lack of theoretical guarantees is a significant curb on the utilization of deep learning approaches.

A recent work [Combettes and Pesquet, 2018] provides a theoretical framework which enables to evaluate the robustness of a network. In this section, we will focus on a subclass of problem (4.2) where both $f(\cdot, \boldsymbol{y})$ and $\mathcal{R}$ are quadratic functions. After highlighting the similarities between the proposed architecture and generic feedforward networks in that case, we will give explicit conditions under which the robustness of the proposed architecture is ensured.

4.5.1 Relation to generic deep neural networks

Although the proposed architecture may seem specific to Algorithm 14, it is actually very similar to generic feedforward neural networks, which are presented in Section 2.1.4.1. For the sake of simplicity, we will consider the variational problem,

$$\underset{\boldsymbol{x}\in\mathcal{C}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{H}\boldsymbol{x}-\boldsymbol{y}\|^2 + \frac{\lambda}{2}\|\boldsymbol{D}\boldsymbol{x}\|^2, \tag{4.34}$$

where $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{H} \in \mathbb{R}^{m\times m}$, $\boldsymbol{D} \in \mathbb{R}^{m\times m}$, and $\mathcal{C}$ is defined as in (4.3). Moreover, we assume that no post–processing layer $\mathcal{L}_{\text{pp}}$ is used. Following the notation of Section 4.4, $(\forall k \in \{0,\dots,K-1\})$ $(\mu_k, \gamma_k, \lambda_k)$ are given positive real numbers, K being the number of layers of the network. Then, for every $k \in \{0,\dots,K-1\}$, layer $\mathcal{L}_k$ corresponds to the following update,

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \operatorname{prox}_{\gamma_k\mu_k\mathcal{B}}\Big(\boldsymbol{x}_k - \gamma_k\Big(\boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{x}_k-\boldsymbol{y}) + \lambda_k\boldsymbol{D}^\top\boldsymbol{D}\boldsymbol{x}_k\Big)\Big)\\
&= \operatorname{prox}_{\gamma_k\mu_k\mathcal{B}}\Big(\Big[\boldsymbol{I}_m - \gamma_k\Big(\boldsymbol{H}^\top\boldsymbol{H} + \lambda_k\boldsymbol{D}^\top\boldsymbol{D}\Big)\Big]\boldsymbol{x}_k + \gamma_k\boldsymbol{H}^\top\boldsymbol{y}\Big),
\end{aligned}$$

where $\mathcal{B}$ is defined as in (4.5). For every $k \in \{0,\dots,K-1\}$, we set

$$\begin{aligned}
\boldsymbol{W}_k &= \boldsymbol{I}_m - \gamma_k\left(\boldsymbol{H}^\top\boldsymbol{H} + \lambda_k\boldsymbol{D}^\top\boldsymbol{D}\right),\\
\boldsymbol{b}_k &= \gamma_k\boldsymbol{H}^\top\boldsymbol{y},\\
R_k &= \operatorname{prox}_{\gamma_k\mu_k\mathcal{B}}.
\end{aligned} \tag{4.35}$$

Then, the K-layer network $\mathcal{L}_{K-1}\circ\cdots\circ\mathcal{L}_0$ is equivalent to (2.7):

$$R_{K-1}\circ(\boldsymbol{W}_{K-1}\cdot+\boldsymbol{b}_{K-1})\circ\cdots\circ R_0\circ(\boldsymbol{W}_0\cdot+\boldsymbol{b}_0),$$

where $(\boldsymbol{W}_k)_{0\le k\le K-1}$ and $(\boldsymbol{b}_k)_{0\le k\le K-1}$ are interpreted as weight operators and bias parameters, respectively. The operators $(R_k)_{0\le k\le K-1}$ defined in

(4.35) can be viewed as specific activation functions since, as shown in [Combettes and Pesquet, 2018], every standard activation function can be derived from a proximity operator. In addition, using [Bauschke and Combettes, 2017, Proposition 24.8(iii)], for every $k \in \{0, \ldots, K-1\}$, R_k can be re-written as the sum of a *proximal activation operator* [Combettes and Pesquet, 2018, Definition 2.20] and a bias.

4.5.2 Preliminary results

Before stating our main stability theorem, we recall the result from [Combettes and Pesquet, 2018, Lemma 3.3] in Proposition 4.5.1 below. We then derive Proposition 4.5.2, which will appear useful when addressing the robustness of the global network. Here, $\mathfrak{S}_m$ denotes the set of symmetric matrices in $\mathbb{R}^{m\times m}$.

Proposition 4.5.1 [Combettes and Pesquet, 2018] *Let $K \geq 1$ be an integer and set $\theta_{-1} = 1$. For every $k \in \{0, \ldots, K-1\}$, let $\boldsymbol{W}_k \in \mathbb{R}^{m\times m}$ and let θ_k be defined by*

$$\theta_k = |||\boldsymbol{W}_k \circ \cdots \circ \boldsymbol{W}_0||| + \sum_{\ell=0}^{k-1} \sum_{0\leq j_0<\cdots<j_\ell\leq k-1} |||\boldsymbol{W}_k \circ \cdots \circ \boldsymbol{W}_{j_\ell+1}||| \times |||\boldsymbol{W}_{j_\ell} \circ \cdots \circ \boldsymbol{W}_{j_{\ell-1}+1}||| \cdots |||\boldsymbol{W}_{j_0} \circ \cdots \circ \boldsymbol{W}_0|||.$$

Then, for every $k \in \{0, \ldots, K-1\}$, $\theta_k = \sum_{\ell=0}^{k} \theta_{\ell-1} |||\boldsymbol{W}_k \circ \ldots \circ \boldsymbol{W}_l|||$.

Proposition 4.5.2 *Let $K \in \mathbb{N}^*$, $\theta \in \mathbb{R}_+^*$, and $\alpha \in [1/2, 1]$. Let $\boldsymbol{W} \in \mathfrak{S}_m$ and let β_- and β_+ denote the smallest and largest eigenvalues of $\boldsymbol{W}$, respectively. Then, the condition*

$$|||\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m||| - |||\boldsymbol{W}||| + 2\theta \leq 2^K\alpha \tag{4.36}$$

is satisfied if and only if one of the following conditions holds:

(i) $\beta_+ + \beta_- \leq 0$ *and* $\theta \leq 2^{K-1}(2\alpha - 1)$;

(ii) $0 \leq \beta_+ + \beta_- \leq 2^{K+1}(1-\alpha)$ *and* $2\theta \leq \beta_+ + \beta_- + 2^K(2\alpha - 1)$;

(iii) $2^{K+1}(1-\alpha) \leq \beta_+ + \beta_-$ *and* $\theta \leq 2^{K-1}$.

Proof. Let $\alpha \in [1/2, 1]$. Since $\boldsymbol{W} \in \mathfrak{S}_m$, we have, $|||\boldsymbol{W}||| = \max\{\beta_+, -\beta_-\}$, and

$$|||\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m||| = \max\left\{\beta_+ - 2^K(1-\alpha), -\beta_- + 2^K(1-\alpha)\right\}. \tag{4.37}$$

Three different cases arise that we review below.

(i) If $\beta_+ + \beta_- \leq 0$ then $|||\boldsymbol{W}||| = -\beta_-$ and

$$\beta_+ - 2^K(1-\alpha) \leq -\beta_- + 2^K(1-\alpha). \tag{4.38}$$

From (4.37) and (4.38), we deduce that $\left|\left|\left|\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m\right|\right|\right| = -\beta_- + 2^K(1-\alpha)$. Replacing $|||\boldsymbol{W}|||$ and $\left|\left|\left|\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m\right|\right|\right|$ by their value in (4.36) leads to Proposition 4.5.2(i).

(ii) If $0 \leq \beta_+ + \beta_- \leq 2^{K+1}(1-\alpha)$ then $|||\boldsymbol{W}||| = \beta_+$ and (4.38) is satisfied. Hence, $\left|\left|\left|\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m\right|\right|\right| = -\beta_- + 2^K(1-\alpha)$. Replacing $|||\boldsymbol{W}|||$ and $\left|\left|\left|\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m\right|\right|\right|$ by their value in (4.36) leads to Proposition 4.5.2(ii).

(iii) If $2^{K+1}(1-\alpha) \leq \beta_+ + \beta_-$ then $|||\boldsymbol{W}||| = \beta_+$ and

$$\beta_+ - 2^K(1-\alpha) \geq -\beta_- + 2^K(1-\alpha). \tag{4.39}$$

From (4.37) and (4.39), we deduce that $\left|\left|\left|\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m\right|\right|\right| = \beta_+ - 2^K(1-\alpha)$. Replacing $|||\boldsymbol{W}|||$ and $\left|\left|\left|\boldsymbol{W} - 2^K(1-\alpha)\boldsymbol{I}_m\right|\right|\right|$ by their value in (4.36) leads to Proposition 4.5.2(iii), which completes the proof.

□

4.5.3 Robustness of iRestNet to an input perturbation

Nonexpansiveness can be used to measure the stability of an operator. For instance, let $T : \mathbb{R}^m \to \mathbb{R}^m$ be a nonexpansive operator and let $\boldsymbol{x}$ and $\boldsymbol{\delta_x}$ be in $\mathbb{R}^m$, where $\boldsymbol{\delta_x}$ can be seen as an input perturbation. From the definition of a nonexpansive operator (Section 2.2.1.4), we deduce that the distance between the perturbed and the original outputs is bounded by the norm of the perturbation; namely, $\|\Delta T(\boldsymbol{x})\| \leq \|\boldsymbol{\delta_x}\|$, with $\Delta T(\boldsymbol{x}) = T(\boldsymbol{x}+\boldsymbol{\delta_x}) - T(\boldsymbol{x})$.

An even stronger indicator of robustness is given by the notion of averageness. If we assume that T is an averaged operator, then we deduce from Proposition 2.2.18 that, as $\|\boldsymbol{\delta}_x\|$ tends to 0, $\Delta T(\boldsymbol{x})$ tends to $\boldsymbol{\delta_x}$. Hence, the notion of averageness, in addition to providing a tighter bound on the norm of the ouput perturbation, also provides an insight regarding its direction.

Let us consider problem (4.34), where we assume additionally that $\boldsymbol{H}^\top\boldsymbol{H}$ and $\boldsymbol{D}^\top\boldsymbol{D}$ are diagonalizable in a same basis denoted $\mathcal{P}$. The latter is satisfied for instance if $\boldsymbol{H}$ and $\boldsymbol{D}$ are the results of cyclic convolutive operators. Theorem 4.5.3 below gives sufficient conditions under which the proposed network applied to problem (4.34) is averaged.

Theorem 4.5.3 *Let $\alpha \in [1/2, 1]$, $(\boldsymbol{W}_k, \boldsymbol{b}_k, R_k)_{0\leq k\leq K-1}$ be defined by (4.35), and $(\theta_k)_{-1\leq k\leq K-1}$ be defined as in Proposition 4.5.1. Let β_- and β_+ be the smallest and largest eigenvalues of $\boldsymbol{W} = \boldsymbol{W}_{K-1} \circ \cdots \circ \boldsymbol{W}_0$, respectively. For every $p \in \{1, \ldots, m\}$ and every $k \in \{0, \ldots, K-1\}$, let $\beta_k^{(p)} = 1 - \gamma_k \left(\beta_{\boldsymbol{H}}^{(p)} + \lambda_k \beta_{\boldsymbol{D}}^{(p)}\right)$, where $\beta_{\boldsymbol{H}}^{(p)}$ and $\beta_{\boldsymbol{D}}^{(p)}$ denote the pth eigenvalue of $\boldsymbol{H}^\top \boldsymbol{H}$ and $\boldsymbol{D}^\top \boldsymbol{D}$ in $\mathcal{P}$, respectively. Then, β_-, β_+, and $(\forall k \in \{0, \ldots, K-1\})$ θ_k can be computed as follows:*

$$\beta_- = \min_{1\leq p\leq m} \prod_{k=0}^{K-1} \beta_k^{(p)}, \qquad \beta_+ = \max_{1\leq p\leq m} \prod_{k=0}^{K-1} \beta_k^{(p)}, \tag{4.40}$$

and

$$\theta_k = \sum_{l=0}^{k} \theta_{l-1} \max_{1\leq q_l\leq m} \left|\beta_k^{(q_l)} \ldots \beta_l^{(q_l)}\right|. \tag{4.41}$$

In addition, if one of the following conditions is satisfied

(i) $\beta_+ + \beta_- \leq 0$ *and* $\theta_{K-1} \leq 2^{K-1}(2\alpha - 1)$;

(ii) $0 \leq \beta_+ + \beta_- \leq 2^{K+1}(1-\alpha)$ *and* $2\theta_{K-1} \leq \beta_+ + \beta_- + 2^K(2\alpha - 1)$;

(iii) $2^{K+1}(1-\alpha) \leq \beta_+ + \beta_-$ *and* $\theta_{K-1} \leq 2^{K-1}$,

then the operator $R_{K-1} \circ (\boldsymbol{W}_{K-1} \cdot + \boldsymbol{b}_{K-1}) \circ \cdots \circ R_0 \circ (\boldsymbol{W}_0 \cdot + \boldsymbol{b}_0)$ is α–averaged.

Proof. If $\boldsymbol{H}^\top \boldsymbol{H}$ and $\boldsymbol{D}^\top \boldsymbol{D}$ are diagonalizable in the same basis then $\boldsymbol{W} \in \mathfrak{S}_m$, which, combined with Proposition 4.5.1, leads to (4.40) and (4.41). If one of the conditions (i)–(iii) is satisfied, then we deduce from Proposition 4.5.2 that $\boldsymbol{W}$ satisfies [Combettes and Pesquet, 2018, Proposition 3.6(iii)] and [Combettes and Pesquet, 2018, Condition 3.1]. In addition, for every $k \in \{0, \ldots, K-1\}$, $R_k(\cdot + \boldsymbol{b}_k)$ is firmly nonexpansive [Bauschke and Combettes, 2017, Proposition 12.28]. Finally, [Combettes and Pesquet, 2018, Theorem 3.8] completes the proof. □

The conditions provided by Theorem 4.5.3 can be easily checked using (4.40). Theorem 4.5.3 provides a framework under which iRestNet is robust to a perturbation of its input: the upper bound of the output perturbation can then be derived from Proposition 2.2.18.

§ 4.6 Numerical experiments

In this section, we present numerical experiments on a set of problems of image restoration, demonstrating that in many cases the proposed approach

yields a better reconstruction quality than standard variational and machine learning methods.

4.6.1 Image deblurring

We consider the non-blind color image deblurring problem, whose degradation model reads

$$\boldsymbol{y} = \boldsymbol{H}\overline{\boldsymbol{x}} + \boldsymbol{\omega},$$

where n is the number of pixels, $\boldsymbol{y} = (\boldsymbol{y}^{(j)})_{1\leq j\leq 3} \in \mathbb{R}^{3n}$ is the blurred RGB image, $\overline{\boldsymbol{x}} = (\overline{\boldsymbol{x}}^{(j)})_{1\leq j\leq 3} \in \mathbb{R}^{3n}$ is the ground-truth, $\boldsymbol{H} \in R^{3n\times 3n}$ is a linear operator that models the circular convolution of a known blur kernel with each channel of the color image, and $\boldsymbol{\omega} \in \mathbb{R}^{3n}$ is a realization of an additive white Gaussian noise with standard deviation σ. An estimate of $\overline{\boldsymbol{x}}$ can be derived from the following penalized formulation, which includes a smoothed total variation regularization,

$$\underset{\boldsymbol{x}\in\mathcal{C}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|^2 + \lambda \sum_{i=1}^{3n} \sqrt{\frac{(\boldsymbol{D}_{\text{v}}\boldsymbol{x})_i^2 + (\boldsymbol{D}_{\text{h}}\boldsymbol{x})_i^2}{\delta^2} + 1}, \tag{4.42}$$

where the feasible set $\mathcal{C}$ is the hypercube $[x_{\min}, x_{\max}]^{3n}$, $x_{\min}$ and $x_{\max}$ are a lower and an upper bound on the pixel intensity, respectively, $\boldsymbol{D}_{\text{v}} \in \mathbb{R}^{3n\times 3n}$ and $\boldsymbol{D}_{\text{h}} \in \mathbb{R}^{3n\times 3n}$ are the vertical and horizontal gradient operators, respectively, $\delta \in \mathbb{R}_+^*$ is a smoothing parameter and $\lambda \in \mathbb{R}_+$ is the regularization parameter. Here, $x_{\min} = 0$, $x_{\max} = 1$ and we set $\delta = 0.01$ in all experiments, which appears as an appropriate order of magnitude. To find this value for δ, we solved problem (4.42) for a small set of images of the database and used the simplex method to find the best values for δ and λ in terms of image quality. It is worth noting that the value for δ has not been fine-tuned, but that the proposed architecture could also be easily modified to include the inference of δ. The update $\mathcal{A}$, defined in (4.32), is derived from (4.42), and is unfolded over K iterations, as it is described in Section 4.4. The bound constraints in problem (4.42) fall under the framework studied in Section 4.3.2, which provides the expression for the proximity operator of the barrier and its gradient.

4.6.2 Network characteristics

The tuning of the number of unfolded iterations K must achieve a compromise between training time, memory requirement, and performance. In order to determine a suitable setting for K, we trained networks with different numbers of layers and increased the number of layers until the performance of the network did not improve significantly. Using this procedure, the depth of iRestNet is taken equal to $K = 40$.

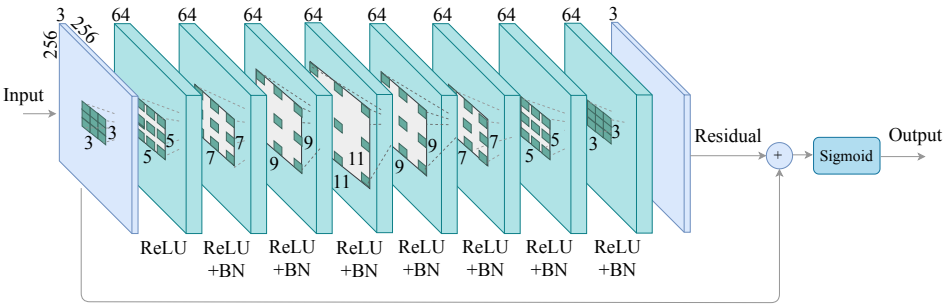


Figure 4.4: *Architecture of* $\mathcal{L}_{\text{pp}}$. *BN: batch normalization.*

Regarding the hidden structures $(\mathcal{L}_k^{(\lambda)})_{0\leq k\leq K-1}$, which estimate the regularization parameter, they are chosen in view of the regularization function used in problem (4.42) and have the following expression,

$$(\forall k \in \{0,\dots,K-1\}) \quad \lambda_k = \mathcal{L}_k^{(\lambda)}(\boldsymbol{x}_k) = \frac{\text{Softplus}(b_k)\,\widehat{\sigma}(\boldsymbol{y})}{\eta(\boldsymbol{x}_k) + \text{Softplus}(c_k)}, \tag{4.43}$$

where (b_k, c_k) is a pair of scalars learned by the network, $\eta(\boldsymbol{x}_k)$ is the standard deviation of $[(\boldsymbol{D}_{\text{v}}\boldsymbol{x}_k)^\top (\boldsymbol{D}_{\text{h}}\boldsymbol{x}_k)^\top]$, which are the concatenated spatial gradients of $\boldsymbol{x}_k$, and $\widehat{\sigma}(\boldsymbol{y})$ is an approximation of the noise level in the blurred image. The noise level is estimated as in [Mallat, 1999, Section 11.3.1]

$$\widehat{\sigma}(\boldsymbol{y}) = \text{median}(|\boldsymbol{W}_{\text{H}}\boldsymbol{y}|)/0.6745,$$

where $|\boldsymbol{W}_{\text{H}}\boldsymbol{y}|$ is the vector gathering the absolute value of the diagonal coefficients of the first level Haar wavelet decomposition of $\boldsymbol{y}$. It is worth noticing that the proposed architecture does not require any prior knowledge about the noise level, in particular the noise standard deviation does not have to be the same for all input images.

The architecture of the post-processing layer $\mathcal{L}_{\text{pp}}$ is inspired from [Zhang et al., 2017c], it is made of 9 convolutional layers with filters of size 3×3. The dilation factor changes from one layer to another, so as to widen the receptive field without creating memory issues. There is little correlation between the artifacts that remain in the image after going through the 40 blocks of iRestNet and the ground-truth image. Hence, it is easier for the network to learn the residual mapping instead of the image itself. Therefore, we add a skip connection between the input of $\mathcal{L}_{\text{pp}}$ and its output. Finally, a ReLU activation function is used after each convolution, the final activation function is chosen as the Sigmoid function, and residual learning is combined with batch normalization, a technique which is widely used in deep learning to accelerate and stabilize the training process [Zhang et al., 2017c]. The final architecture of $\mathcal{L}_{\text{pp}}$ can be found in Figure 4.4.

4.6.3 Dataset and experimental settings

The training set is made of 1200 RGB images: 200 images stem from the Berkeley segmentation (BSD500) training set, while the remaining 1000 images are taken from the COCO training set. We use the BSD500 validation set, which is made of 100 images, to monitor the training and check if there is overfitting. The performance of the proposed method is evaluated on two different test sets: the BSD500 test set, which is made of 200 RGB images, and the Flickr30 test set used in [Xu et al., 2014a], which is made of 30 RGB images. The test images have been center-cropped using a window of size 256×256. Blurry images are produced using the following 25×25 blur kernels and noise levels:

- A Gaussian kernel, which models atmospheric turbulence, with a standard deviation of 1.6 pixels, and a Gaussian noise standard deviation of $\sigma = 0.008$. This configuration is denoted as GaussianA. To evaluate the robustness of the proposed method with respect to the noise level, the same kernel is used with a Gaussian noise whose standard deviation is uniformly distributed between 0.01 and 0.05. The latter is denoted as GaussianB.

- The Gaussian kernel with a standard deviation of 3 pixels, and a Gaussian noise standard deviation of $\sigma = 0.04$, denoted as GaussianC.

- The eighth and third motion test kernels from [Levin et al., 2009], which are real-world camera shake kernels, with a Gaussian noise standard deviation of $\sigma = 0.01$. These settings are denoted as MotionA and MotionB, respectively.

- The square uniform kernel of size 7×7, with a Gaussian noise standard deviation of $\sigma = 0.01$. This configuration is referred to as Square.

4.6.4 Training

For each degradation model, one iRestNet network is trained. We use a greedy approach for training the first 30 layers. For $\mathcal{L}_0$, a minibatch of 10 images is selected at every iteration, randomly cropped using a window of size 256×256, blurred with the given kernel, and degraded with Gaussian noise; the training of $\mathcal{L}_0$ stops after a fixed number of epochs. Then, for each image of the training set, a random crop of size 256×256 is selected, blurred, corrupted with noise and passed through $\mathcal{L}_0$, the output is saved and used as an input to train $\mathcal{L}_1$. When the training of $\mathcal{L}_1$ is complete, its output is used to train the next layer, etc... This training strategy is chosen with regards to its low memory requirement: the number of layers is not limited by the hardware.

	GaussianA	GaussianB	GaussianC
Rates	(0.01,0.001)	(0.01,0.001)	(0.001,0.001)
Epochs	(40,393)	(40,340)	(40,300)

Table 4.1: *Training information for Gaussian blur kernels: initial learning rates and number of epochs. For every couple, the first and second numbers correspond to the training of* $(\mathcal{L}_k)_{0\leq k\leq 29}$ *and* $\mathcal{L}_{\text{pp}}\circ\mathcal{L}_{39}\circ\ldots\circ\mathcal{L}_{30}$*, respectively.*

	MotionA	MotionB	Square
Rates	(0.01,0.002)	(0.01,0.001)	(0.01,0.005)
Epochs	(40,1200)	(40,1250)	(40,740)

Table 4.2: *Training information for motion and square blur kernels: initial learning rates and number of epochs. For every couple, the first and second numbers correspond to the training of* $(\mathcal{L}_k)_{0\leq k\leq 29}$ *and* $\mathcal{L}_{\text{pp}} \circ \mathcal{L}_{39} \circ \ldots \circ \mathcal{L}_{30}$*, respectively.*

The rest of the network, $\mathcal{L}_{\text{pp}} \circ \mathcal{L}_{39} \circ \ldots \circ \mathcal{L}_{30}$, is trained as one block and the learning rate is multiplied by 0.9 every 50 epochs. To accelerate the training, for every $k \in \{1, \ldots, K-1\}$, the weights of $\mathcal{L}_k$ are initialized with those of $\mathcal{L}_{k-1}$. Detailed information about learning rates and number of epochs can be found in Tables 4.1 and 4.2 below.

The validation set is used to monitor this last step of the training. In particular, the configuration of network parameters that gives the best performance on the validation set during the training is the one saved and used for the tests. Note that for the first 30 layers, after each layer the quality of the restored training images should improve. This property comes from the training strategy, it is not encoded in the network: if memory was not an issue, then iRestNet could be trained in an end-to-end fashion.

We use the Adam optimizer [Kingma and Ba, 2014] to minimize the training loss, which is taken as the negative of the structural similarity measure (SSIM) [Wang et al., 2004] defined below

$$\text{SSIM}(\boldsymbol{x}, \overline{\boldsymbol{x}}) = \frac{(2\mu_x\mu_{\overline{\boldsymbol{x}}} + c_1)(2\sigma_{\boldsymbol{x}}\sigma_{\overline{\boldsymbol{x}}} + c_2)(2\text{cov}_{\boldsymbol{x}\overline{\boldsymbol{x}}} + c_3)}{(\mu_{\boldsymbol{x}}^2 + \mu_{\overline{\boldsymbol{x}}}^2 + c_1)(\sigma_{\boldsymbol{x}}^2 + \sigma_{\overline{\boldsymbol{x}}}^2 + c_2)(\sigma_{\boldsymbol{x}}\sigma_{\overline{\boldsymbol{x}}} + c_3)},$$

where $\overline{\boldsymbol{x}}$ is the ground truth, $\boldsymbol{x}$ is the restored image, $(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}})$ and $(\mu_{\overline{\boldsymbol{x}}}, \sigma_{\overline{\boldsymbol{x}}})$ are mean and standard deviation of $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}$, respectively, $\text{cov}_{\boldsymbol{x}\overline{\boldsymbol{x}}}$ is the cross–covariance of $\boldsymbol{x}$ and $\overline{\boldsymbol{x}}$, and c_1, c_2 and c_3 are constants. As explained in [Wang et al., 2004], the SSIM is a good measure of perceived visual quality, since it is based on how the human eye extracts structural information from an image. Hence, it is more discriminative with regards to artifacts than the mean square error for instance. The gradient of the SSIM loss with

respect to the trainable parameters of the network is computed using a code available online[1] and based on [Wang et al., 2004], the chain rule, automatic differentiation [Paszke et al., 2017], and the expression given in Section 4.3.2 for the derivatives of the barrier proximity operator.

Codes are implemented in Pytorch. Some hidden layers in the post-processing part make use of ReLU, which is not differentiable everywhere. Since this nondifferentiability happens only at specific points for which the left and right derivatives are well–defined, Pytorch can handle it as explained in [Goodfellow et al., 2016]. All trainings are conducted using a GeForce GTX 1080 GPU or a Tesla V100 GPU. The training, which can be performed off-line, takes approximately 3 to 4 days for each blur kernel, while the time taken per test image is only about 1.4 sec on a GeForce GTX 1080 GPU.

4.6.5 Evaluation metrics and competitors

The restoration is evaluated in terms of the SSIM metric. The reconstruction given by the proposed approach is compared with a solution to problem (4.42) obtained using the projected gradient algorithm [Iusem, 2003]. For every blurred image, the pair (λ, δ) which leads to the best SSIM is selected using the simplex method. The solution given by this variational approach is referred to as VAR. The latter is an unrealistic scenario since it assumes that there is a perfect estimator of the error for selecting the hyperparameters, but it gives an upper bound on the image quality that one can expect by solving (4.42). We also use the following deep learning image restoration methods for comparison.

(i) EPLL [Zoran and Weiss, 2011] corresponds to a Bayesian approach where the authors divide images into patches and assign a prior to each patch. This results in a Gaussian mixture model whose parameters are learnt. The deblurred image is then a maximum a posteriori estimate of the image.

(ii) MLP refers to the Multi-Layer Perceptron network proposed in [Schuler et al., 2013], which is fed with a pre-deconvolved image produced by a Wiener deconvolution filter. The network is trained for each blurring kernel.

Finally, we include comparisons with three unfolded-based methods.

(iii) In IRCNN [Zhang et al., 2017c], an empirical algorithm derived from an augmented Lagrangian formulation is unfolded over 30 iterations[2],

[1] https://github.com/Po-Hsun-Su/pytorch-ssim

[2] In [Zhang et al., 2017c], this algorithm is improperly called half-quadratic splitting, but it does not correspond to usual half-quadratic optimization methods described for instance in [Allain et al., 2006]. Actually, the algorithm unfolded in [Zhang et al., 2017c] can be interpreted as a preconditioned forward-backward algorithm.

and a CNN is used as a denoiser to update the splitting variable.

(iv) In FCNN [Zhang et al., 2017a], the authors unfold the same algorithm as in the previous reference, and use a network to learn an effective regularization function.

(v) The method from [Meinhardt et al., 2017] is referred to as PDHG. The authors perform a maximum of 30 iterations of a primal dual hybrid gradient algorithm, and the proximity operator of the second regularization function is replaced by a neural network.

For FCNN, we use the code that is available online, in which the authors provide a model that has only been trained for motion blurs. Hence, for a fair comparison, we only provide the results of FCNN on MotionA and MotionB, and we specify that this method is not applicable (n/a) to the other configurations. Similarly, for MLP and PDHG, the authors do not provide models that were trained specifically for MotionB and Square, so we do not test these methods on these two configurations.

Since MLP, EPLL and IRCNN require the knowledge of the noise level, for the GaussianB degradation model, we make use of the estimation of the noise standard deviation given by the method in [Mallat, 1999, Section 11.3.1]. In addition, since some comparison methods, like EPLL for instance, do not estimate well the borders of the images, the SSIM index is computed excluding a 6-pixel-wide frame for all images and all tested methods.

4.6.6 Results and discussion

The average SSIM obtained with the different methods for the various blur kernels and noise levels on the BSD500 test set can be found in Table 4.3. The mean SSIM achieved with iRestNet on this test set is greater than those obtained with the other methods for all degradation models except MotionA. For this kernel, the average SSIM achieved with iRestNet is the second highest value after IRCNN, which appears as the most competitive method. IRCNN involves two steps: first, a Wiener filter is applied to the blurred image, then, a neural network is used to predict the residual and denoise the image. These two steps are repeated 30 times, for 30 different manually tuned regularization parameters. In contrast, iRestNet does not require any tuning from the user regarding the regularization parameters during training. For completeness, the SSIM of all images of the BSD500 test set are plotted in Figure 4.5 for the 6 different degradation models. As one can see, iRestNet performs well in terms of SSIM on most of the images.

Since no image was taken from Flickr for training iRestNet, the results on the Flickr30 test set show how well the performance of the trained networks are transferable on test sets with statistics that are different from those of

	GaussianA	GaussianB	GaussianC	MotionA	MotionB	Square
Blurred	0.676	0.526	0.326	0.383	0.549	0.544
VAR	0.804	0.723	0.587	0.819	0.829	0.756
EPLL	0.800	0.708	0.565	0.816	0.839	0.755
MLP	0.821	0.734	0.608	0.854	n/a	n/a
PDHG	0.796	0.716	0.563	0.801	n/a	n/a
IRCNN	0.841	0.768	0.619	**0.902**	0.907	0.834
FCNN	n/a	n/a	n/a	0.794	0.847	n/a
iRestNet	**0.853**	**0.787**	**0.641**	0.898	**0.910**	**0.840**

Table 4.3: *SSIM results on the BSD500 test set.*

	GaussianA	GaussianB	GaussianC	MotionA	MotionB	Square
Blurred	0.723	0.545	0.355	0.376	0.590	0.579
VAR	0.857	0.776	0.639	0.856	0.869	0.818
EPLL	0.860	0.770	0.616	0.857	0.887	0.827
MLP	0.874	0.798	0.668	0.891	n/a	n/a
PDHG	0.853	0.781	0.623	0.855	n/a	n/a
IRCNN	0.885	0.819	0.676	**0.927**	**0.930**	**0.886**
FCNN	n/a	n/a	n/a	0.801	0.890	n/a
iRestNet	**0.892**	**0.833**	**0.696**	0.919	**0.930**	**0.886**

Table 4.4: *SSIM results on the Flickr30 test set.*

the training set. Table 4.4 contains the average SSIM obtained with the different methods on the Flickr30 test set. Similarly to the BSD500 test set, iRestNet compares favorably with the other approaches on the Flickr30 test set.

Examples of visual results obtained with the different methods can be found in Figures 4.6 and 4.7 for two images from the BSD500 test set and the blur kernels GaussianB and Square, respectively. We also provide the results obtained for one image from the Flickr30 test set that has been degraded with MotionB. As one can see from inspecting these pictures, details from the snake's and caterpillar's skin patterns are better retrieved with iRestNet, which provides more visually-satisfactory results than competitors. Similarly, on Figure 4.8, competitors tend to smooth too much the details on the leaves as it can be seen in the top left-hand corner. Regarding Figure 4.6, which belongs to the test set with a level-varying noise, it is worth noting that, on the result obtained with the proposed method, the green background is free from artifacts, which is not the case for the other methods, in particular for PDHG and IRCNN. This suggests that those two competitors are not robust to a small change in the noise level.

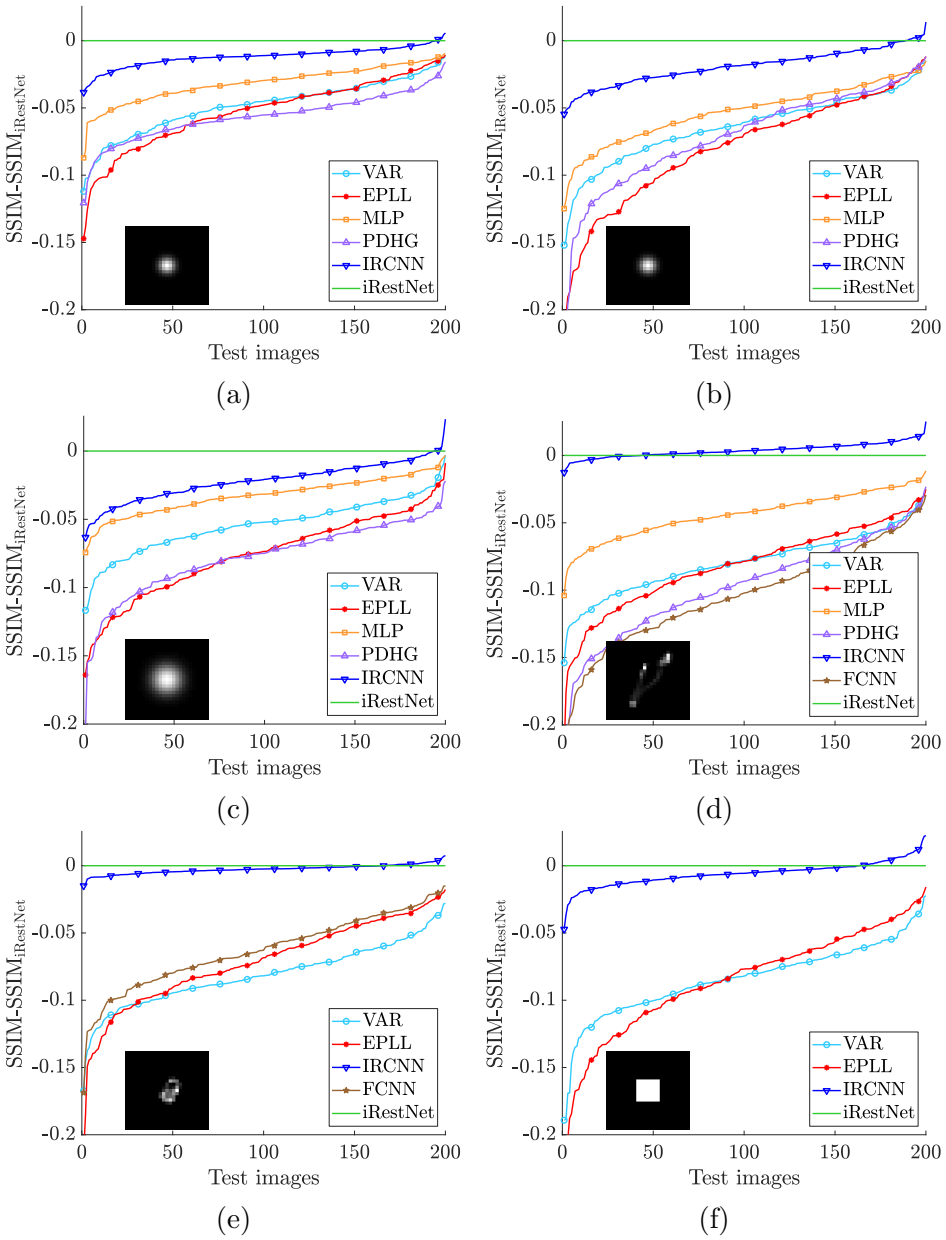


Figure 4.5: *Sorted improvement of iRestNet with regards to other methods on the BSD500 test set using the SSIM metric: a negative value indicates a better performance of iRestNet. (a): GaussianA, (b): GaussianB, (c): GaussianC, (d): MotionA, (e): MotionB, (f): Square.*

Figure 4.9 shows the stepsize, barrier parameter and regularization weight sequences obtained by passing the image from Figure 4.6 through the 40 layers of iRestNet.

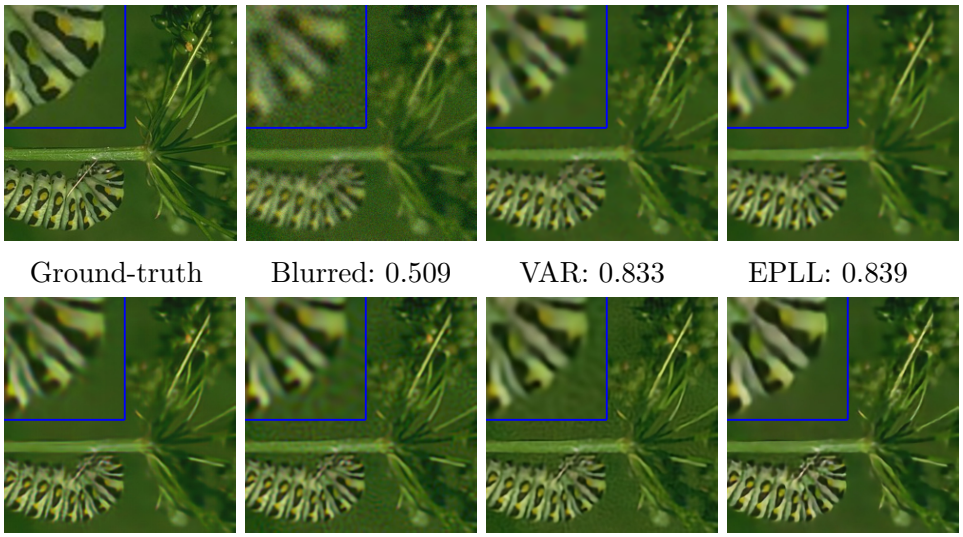


Figure 4.6: *Visual results and SSIM obtained with the different methods on one image from the BSD500 test set degraded with GaussianB.*

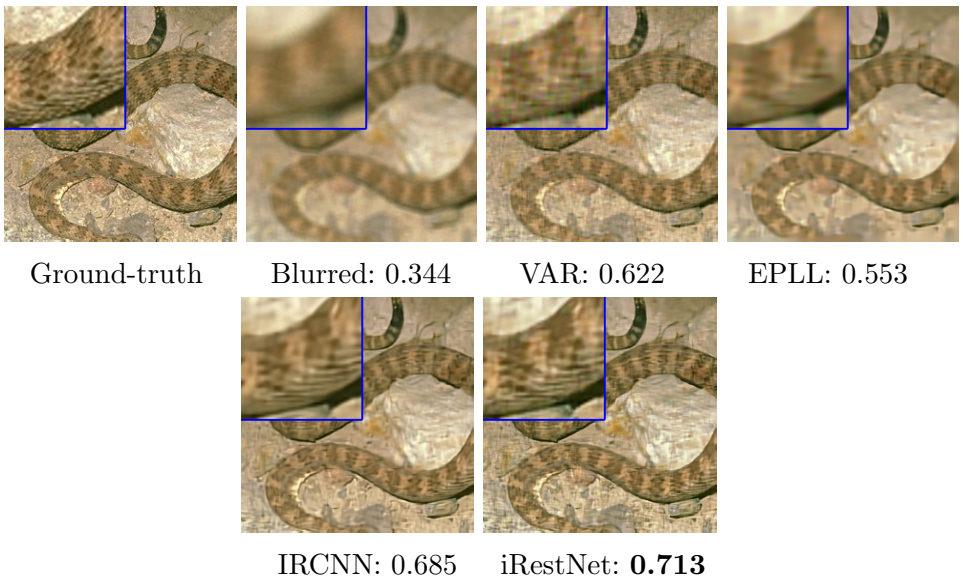


Figure 4.7: *Visual results and SSIM obtained with the different methods on one image from the BSD500 test set degraded with Square.*

§ 4.7 Summary

From a variational formulation of an inverse problem, we have derived in this chapter a novel neural network architecture by unfolding a proximal

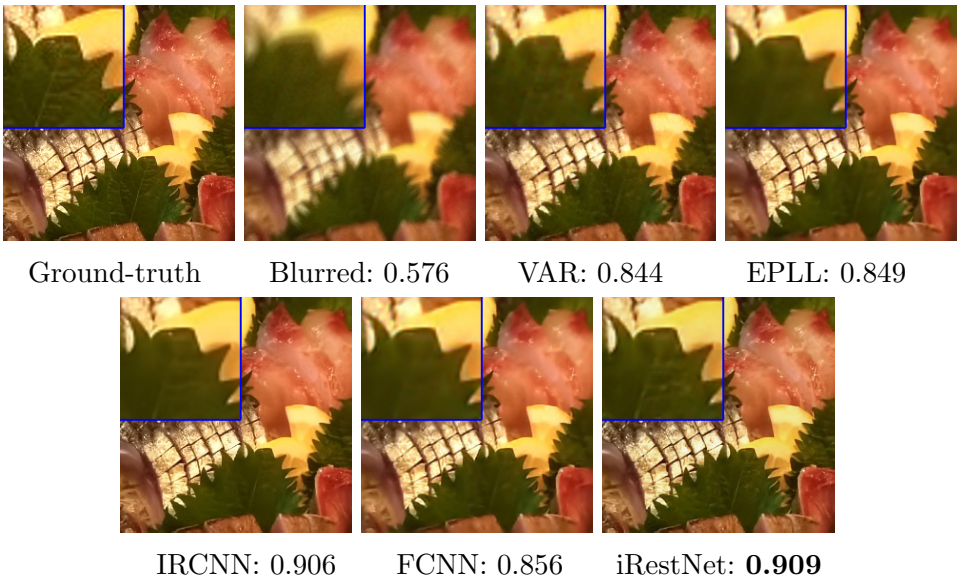


Figure 4.8: *Visual results and SSIM obtained with the different methods on one image from the Flickr30 test set degraded with MotionB.*

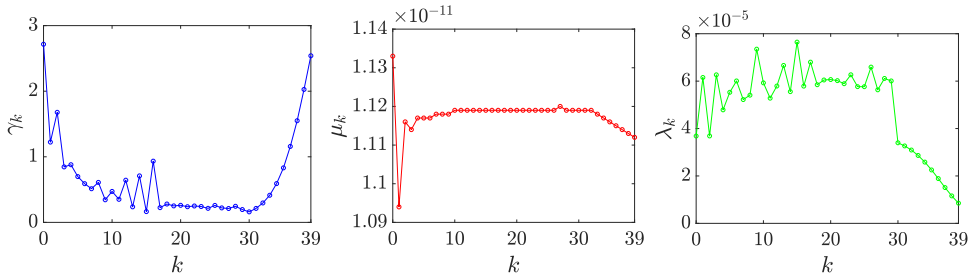


Figure 4.9: *Left to right: estimated stepsize* $(\gamma_k)_{0\leq k\leq K-1}$*, barrier parameter* $(\mu_k)_{0\leq k\leq K-1}$ *and regularization weight* $(\lambda_k)_{0\leq k\leq K-1}$ *for the image from Figure 4.6 passed through the network layers.*

interior point algorithm. It can be noted that the proposed approach can be extended to a set of regularization functions, or to penalizations which are parametrized by several variables. Useful constraints on the sought solution can be enforced thanks to a logarithmic barrier, so providing more control over the output of the network. We have shown for three standard types of constraints that the involved proximity operator can easily be computed, and that its derivatives are well-defined and computable. In the case of a quadratic cost function, the theoretical result of Section 4.5 regarding the robustness of the network with respect to an input perturbation, ensures the reliability of the proposed method, which is crucial for many applications. It would be interesting to extend the scope of this study to a wider class of

problems, and to illustrate this stability result by numerical experiments on different applications like classification. As demonstrated by our experiments in image restoration, iRestNet performs favorably compared to state-of-the-art variational and machine learning methods. An advantage of the proposed approach is that, in contrast with its evaluated competitors, it does not require any knowledge about the noise level and it does not involve any hand-selection of the regularization parameters. One limitation of iRestNet is that the network needs to be trained for a given blur kernel. A direction for future works is to extend the method to situations in which the observation model is not fully known, so as to address blind or semi-blind deconvolution problems.

The unfolded approach proposed in this chapter overcomes a limitation of classical variational strategies, which is the setting of their hyperparameters. Another alternative to address this issue is to use the Bayesian framework, which leads to useful stochastic simulation methods. In addition, similarly to iRestNet whose training is based on a better indicator of the error than the original objective function, Bayesian approaches allow to derive insightful estimators other than the MAP, like the MMSE for instance (please refer to Section 2.1.3 for more details). In the next chapter, we propose to incorporate a proximal algorithm in a Bayesian approach.

- Chapter 5 -

A preconditioned proximal sampling algorithm with applications in ultrasound imaging

Ultrasound (US) imaging is a non-invasive procedure which is widely used in medical applications, such as obstetrics or cardiology. Although the quasi absense of side-effects caused by this procedure, and its low cost and aquisition time, make it the norm for numerous diagnostic exams [Szabo, 2004], ultrasonic measurements suffer from some downsides, including a low signal-to-noise ratio, attenutation, speckle, shadows and a limited resolution [Noble and Boukerroui, 2006; Ploquin et al., 2015].

The echo signals recorded during an ultrasound exam are processed to produce radio-frequency (RF) images. While image processing tasks are commonly performed directly on the resulting RF image [Nair et al., 2002], or on its B-mode [Noble and Boukerroui, 2006], which is a log-compression of its envelope, useful information can also be drawn from the statistics of the scatterer field, also called *tissue reflectivity function* (TRF) [Jensen, 1991; Ng et al., 2006]. Since the TRF models well tissue inhomogeneity, it can be used to perform efficiently segmentation [Pereyra et al., 2012], tissue characterization [Bernard et al., 2006], or classification [Alessandrini et al., 2011]. Estimating the TRF given the RF image is a challenging inverse problem, which we are going to tackle jointly with the segmentation task in this chapter.

Models that are studied in US imaging are derived from the physical laws governing the involved aquisition process. Such models may be significantly complex depending of the task to be performed. The Bayesian framework and stochastic simulation methods, like the ones described in Section 2.1.3, have proven to be very useful for solving inverse problems arising in US imaging [Comer and Delp, 2000; Husby and Rue, 2004; Dobigeon et al., 2012]. In the context of Bayesian theory, we propose in this chapter a preconditioned proximal sampling algorithm for the TRF, which is incorporated in a hybrid Gibbs sampler with the aim of performing joint deconvolution and

segmentation of US images.

This chapter is organized as follows. After formulating the studied inverse problem in Section 5.1 and motivating our approach in Section 5.2, we detail the proposed strategy used to sample the TRF in Section 5.3. Section 5.4 then describes the investigated hierarchical Bayesian model, while the proposed hybrid Gibbs sampler is detailed in Section 5.5. Finally, numerical experiments on both simulated and real US data are presented in Section 5.6 and some conclusions are drawn in Section 5.7.

§ 5.1 Problem statement

Let $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^m$ be the vectorized TRF and radio-frequency image, respectively. The following simplified model is used [Jensen et al., 1993; Ng et al., 2006]

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\omega}, \tag{5.1}$$

where $\boldsymbol{H} \in \mathbb{R}^{m \times m}$ is a linear operator that models the convolution with the point spread function (PSF) of the probe, and $\boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}_m, \sigma^2 \boldsymbol{I}_m)$. In this chapter, we assume that the PSF is known, while the variance of the Gaussian noise $\sigma^2 \in \mathbb{R}_+^*$ is an unknown parameter to be estimated. The TRF is comprised of K different tissues, which are identified by a hidden label field $\boldsymbol{z} = (z_i)_{1 \leq i \leq m} \in \{1, \ldots, K\}^m$. For every $k \in \{1, \ldots, K\}$, the kth region is modeled by a generalized Gaussian distribution ($\mathcal{GGD}$) [Alessandrini et al., 2011; Zhao et al., 2016], which is parametrized by a shape parameter $\alpha_k \in [0, 3]$, related to the scatterer concentration, and a scale parameter $\beta_k \in \mathbb{R}_+^*$, linked to the signal energy. Given $\boldsymbol{y}$ and $\boldsymbol{H}$, the aim is to estimate a deblurred image $\boldsymbol{x}$ [Jensen, 1992; Michailovich and Tannenbaum, 2007], as well as the noise variance σ^2, the $\mathcal{GGD}$ parameters $\boldsymbol{\alpha} = (\alpha_k)_{1 \leq k \leq K}$ and $\boldsymbol{\beta} = (\beta_k)_{1 \leq k \leq K}$, and the label field $\boldsymbol{z}$.

§ 5.2 Motivation

Due to the interdependence of the unknowns introduced in the previous section, it is beneficial to perform the deconvolution and segmentation tasks in a joint manner [Ayasso and Mohammad-Djafari, 2010; Pirayre et al., 2017]. This is achieved in [Zhao et al., 2016] by considering a hierarchical Bayesian model, which is used within an MCMC method[1] to sample $\boldsymbol{x}$, σ^2, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{z}$ according to the full conditional distribution. Despite promising results in image restoration and segmentation, the method in [Zhao et al., 2016] is of significant computational complexity, in particular due to the adjusted

[1] Please refer to Section 2.1.3.2 for more information about MCMC methods.

Hamiltonian Monte Carlo (HMC) method [Neal, 2011; Robert et al., 2018] used to sample the TRF.

Recently, efficient and reliable stochastic sampling strategies have been devised [Durmus et al., 2018; Pereyra, 2016; Schreck et al., 2016] using the proximity operator presented in Section 2.2.1.3, which is known as a useful tool for large-scale nonsmooth optimization [Combettes and Pesquet, 2011]. In this work, we investigate an MCMC algorithm to perform the joint deconvolution and segmentation of US images, where the TRF is sampled with a scheme inspired from the proximal unadjusted Langevin algorithm (P-ULA) [Pereyra, 2016]. P-ULA generates samples according to an approximation of the target distribution without acceptance test, while being geometrically ergodic, whereas classical unadjusted Langevin algorithms may have convergence issues.

In the next section, we propose an original accelerated preconditioned version of P-ULA (PP-ULA), which relies on the use of a variable metric forward-backward strategy [Stuart et al., 2004; Chouzenoux et al., 2014b]. We also introduce in Section 5.5.2 an efficient solver based on the majorize-minimize (MM) principle to tackle the involved non log-concave priors. The proposed PP-ULA is then incorporated within a new hybrid Gibbs sampler summarized in Algorithm 17, yielding a substantial reduction of the computational time needed to perform joint high-quality deconvolution and segmentation of US images.

§ 5.3 Preconditioned P-ULA sampling algorithm

In this section, after reminding results about the Langevin diffusion and its discretization using Euler's scheme, we provide details about the derivation of the proposed sampling algorithm, called PP-ULA. This algorithm will be used to sample the TRF in Section 5.5.

5.3.1 Discrete Langevin diffusion

An m-dimensional Langevin diffusion is a continuous time Markov process $x : \mathbb{R}_+ \to \mathbb{R}^m$, which is the solution to the following stochastic differential equation [Roberts and Stramer, 2002],

$$(\forall t \in \mathbb{R}_+) \quad \mathrm{d}x(t) = b(x(t))\mathrm{d}t + V(x(t))\mathrm{d}B(t), \tag{5.2}$$

where $B : \mathbb{R}_+ \to \mathbb{R}^m$ is a Brownian motion, $V : \mathbb{R}^m \to \mathbb{R}^{m\times m}$ is the volatility matrix-valued function, and the drift term $b : \mathbb{R}^m \to \mathbb{R}^m$ is such that, for

every $\boldsymbol{x} \in \mathbb{R}^m$

$$(\forall i \in \{1,\dots,m\}) \quad (b(\boldsymbol{x}))_i = \frac{1}{2}\sum_{j=1}^{m}\left[(A(\boldsymbol{x}))_{i,j}\frac{\partial \log \pi(\boldsymbol{x})}{\partial x_j} + \det(A(\boldsymbol{x}))^{\frac{1}{2}}\sum_{k=1}^{m}\frac{\partial}{\partial x_j}\left((A(\boldsymbol{x}))_{i,k}\det(A(\boldsymbol{x}))^{-\frac{1}{2}}\right)\right].$$

In the above definition, $(\forall \boldsymbol{x} \in \mathbb{R}^m)\ A(\boldsymbol{x}) = V(\boldsymbol{x})V(\boldsymbol{x})^\top \in \mathbb{R}^{m\times m}$ is a symmetric positive definite matrix, and $\det(A(\boldsymbol{x}))$ denotes its determinant. In addition, π is the density of the stationary distribution of the diffusion. Euler's discretization scheme can be used on (5.2) to generate a Langevin Markov chain $(\boldsymbol{x}^{(t)})_{t\in\mathbb{N}}$ as follows

$$(\forall t \in \mathbb{N}) \quad \boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + 2\gamma b(\boldsymbol{x}^{(t)}) + \sqrt{2\gamma}(A(\boldsymbol{x}^{(t)}))^{\frac{1}{2}}\boldsymbol{\omega}^{(t)}.$$

Hereabove, $(\forall t \in \mathbb{N})\ \boldsymbol{\omega}^{(t)}$ is generated using a zero-mean Gaussian distribution $\mathcal{N}(\mathbf{0}_m, \boldsymbol{I}_m)$, and $\gamma \in \mathbb{R}_+^*$ is the discretization stepsize that controls the length of the jumps, while the scale matrix $A(\cdot)$ drives their direction. Instead of letting $A(\cdot)$ be the identity operator as in the standard Metropolis adjusted Langevin algorithm [Roberts and Tweedie, 1996], we follow [Stuart et al., 2004; Marnissi et al., 2018] and use a preconditioning matrix $\boldsymbol{A} \in \mathfrak{S}_m^+$ to accelerate the Langevin scheme, which leads to

$$(\forall t \in \mathbb{N}) \quad \boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \gamma \boldsymbol{A}\nabla \log \pi(\boldsymbol{x}) + \sqrt{2\gamma}\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{\omega}^{(t+1)}. \tag{5.3}$$

5.3.2 Approximation of the target diffusion

Let $f : \mathbb{R}^m \to]-\infty, +\infty]$ and let $g \in \Gamma_0(\mathbb{R}^m)$ be differentiable on $\mathbb{R}^m$ with a Lipschitz-continuous gradient. Assume that the target distribution π satisfies the following relation,

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \pi(\boldsymbol{x}) \propto \exp(-(f+g)(\boldsymbol{x})).$$

Let $\gamma \in \mathbb{R}_+^*$ and $\boldsymbol{M} \in \mathfrak{S}_m^+$. Following [Pereyra, 2016], we replace π by its Moreau approximation $\pi_\gamma^{\boldsymbol{M}}$ defined by

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \pi_\gamma^{\boldsymbol{M}}(\boldsymbol{x}) = \sup_{\boldsymbol{u}\in\mathbb{R}^m} \pi(\boldsymbol{u}) \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{M}}^2}{2\gamma}\right).$$

Note that we dropped the normalization constant and that, for convergence acceleration purposes, we have introduced the preconditioning matrix $\boldsymbol{M}$, which was not included in [Pereyra, 2016]. When $\boldsymbol{M}$ is not specified, the identity matrix is used, i.e. $\boldsymbol{M} = \boldsymbol{I}_m$. Hence, the approximated version of (5.3) reads

$$(\forall t \in \mathbb{N}) \quad \boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \gamma \boldsymbol{A}\nabla \log \pi_\gamma^{\boldsymbol{M}}(\boldsymbol{x}) + \sqrt{2\gamma}\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{\omega}^{(t+1)}. \tag{5.4}$$

We can then deduce the following result when $f+g$ is convex.

Proposition 5.3.1 *For every* $\gamma \in \mathbb{R}_+^*$, $\boldsymbol{M} \in \mathfrak{S}_m^+$ *and* $\boldsymbol{x} \in \mathbb{R}^m$, *if* $(f+g) \in \Gamma_0(\mathbb{R}^m)$, *then we have*

$$\nabla \log \pi_\gamma^{\boldsymbol{M}}(\boldsymbol{x}) = \boldsymbol{M} \frac{\operatorname{prox}_{\gamma(f+g)}^{\boldsymbol{M}}(\boldsymbol{x}) - \boldsymbol{x}}{\gamma}.$$

Proof. By definition of $\pi_\gamma^{\boldsymbol{M}}$, we have

$$(\forall \boldsymbol{x} \in \mathbb{R}^m) \quad \log \pi_\gamma^{\boldsymbol{M}}(\boldsymbol{x}) = -\operatorname{prox}_{\gamma(f+g)}^{\boldsymbol{M}}(\boldsymbol{x}).$$

Hence, applying [Combettes and Wajs, 2005, Lemma 2.5] in the metric induced by $\boldsymbol{M}$ directly leads to the result.
□

From Proposition 5.3.1, (5.4) becomes

$$(\forall t \in \mathbb{N}) \quad \boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \gamma \boldsymbol{A}\boldsymbol{M} \frac{\operatorname{prox}_{\gamma(f+g)}^{\boldsymbol{M}}(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}}{\gamma} + \sqrt{2\gamma}\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{\omega}^{(t+1)}.$$

Finally, we set $\boldsymbol{A} = \boldsymbol{M}^{-1}$, which leads to

$$(\forall t \in \mathbb{N}) \quad \boldsymbol{x}^{(t+1)} = \operatorname{prox}_{\gamma(f+g)}^{\boldsymbol{M}}(\boldsymbol{x}^{(t)}) + \sqrt{2\gamma}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{\omega}^{(t+1)}. \tag{5.5}$$

Next, we introduce a splitting strategy in (5.5).

5.3.3 Forward-backward approximation

By assumption, g is differentiable on $\mathbb{R}^m$ and its gradient ∇g is Lipschitz-continuous. It is worth noting that the computation of the proximity operator of the sum of two functions is generally intractable [Pustelnik and Condat, 2017]. Hence, as suggested in [Pereyra, 2016], we use a first-order Taylor expansion to approximate the proximity operator of $f+g$ and introduce a forward step in the proposed PP-ULA iteration. Let o denotes Landau's notation.[2]

Let $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{u} \in \mathbb{R}^m$. Using

$$g(\boldsymbol{u}) = g(\boldsymbol{x}) + (\boldsymbol{u}-\boldsymbol{x})^\top \nabla g(\boldsymbol{x}) + o(\|\boldsymbol{u}-\boldsymbol{x}\|),$$

we have

$$(f+g)(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{M}}^2 = f(\boldsymbol{u}) + g(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{M}}^2 + (\boldsymbol{u}-\boldsymbol{x})^\top \nabla g(\boldsymbol{x}) + o(\|\boldsymbol{u}-\boldsymbol{x}\|),$$

[2] Following Landau's notation, we will write that $F(\boldsymbol{u}) = o(\|\boldsymbol{u}-\boldsymbol{x}\|)$, where $F : \mathbb{R}^m \to \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^m$, if $F(\boldsymbol{u})/\|\boldsymbol{u}-\boldsymbol{x}\| \to 0$ as $\boldsymbol{u} \to \boldsymbol{x}$.

which can be re-written as

$$(f+g)(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{M}}^2 = f(\boldsymbol{u}) + g(\boldsymbol{x}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}+\gamma\boldsymbol{M}^{-1}\nabla g(\boldsymbol{x})\|_{\boldsymbol{M}}^2 - \frac{\gamma}{2}\|\boldsymbol{M}^{-\frac{1}{2}}\nabla g(\boldsymbol{x})\|^2 + o(\|\boldsymbol{u}-\boldsymbol{x}\|).$$

Hence, the proximity operator of $f+g$ can be expressed as follows,

$$\begin{aligned}\operatorname{prox}_{\gamma(f+g)}^{\boldsymbol{M}}(\boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^m}(f+g)(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{M}}^2 \\ &= \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^m}\Bigg(f(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}+\gamma\boldsymbol{M}^{-1}\nabla g(\boldsymbol{x})\|_{\boldsymbol{M}}^2 \\ &\qquad + o(\|\boldsymbol{u}-\boldsymbol{x}\|)\Bigg).\end{aligned}$$

In addition, we have

$$\operatorname{prox}_{\gamma f}^{\boldsymbol{M}}(\boldsymbol{x}-\gamma\boldsymbol{M}^{-1}\nabla g(\boldsymbol{x})) = \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^m} f(\boldsymbol{u}) + \frac{1}{2\gamma}\|\boldsymbol{u}-\boldsymbol{x}+\gamma\boldsymbol{M}^{-1}\nabla g(\boldsymbol{x})\|_{\boldsymbol{M}}^2.$$

Therefore, when γ is small, $\operatorname{prox}_{\gamma f}^{\boldsymbol{M}}(\boldsymbol{x}-\gamma\boldsymbol{M}^{-1}\nabla g(\boldsymbol{x}))$ is a good approximation of $\operatorname{prox}_{\gamma(f+g)}^{\boldsymbol{M}}(\boldsymbol{x})$. Plugging this preconditioned forward-backward scheme [Combettes and Pesquet, 2011] in (5.5) leads to the proposed sampling method PP-ULA,

$$(\forall t\in\mathbb{N})\quad \boldsymbol{x}^{(t+1)} = \operatorname{prox}_{\gamma f}^{\boldsymbol{M}}(\boldsymbol{x}^{(t)}-\gamma\boldsymbol{M}^{-1}\nabla g(\boldsymbol{x}^{(t)})) + \sqrt{2\gamma}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{\omega}^{(t+1)}. \quad (5.6)$$

Now that we have presented our proposed sampling strategy PP-ULA, we describe in the next section the Bayesian model to which it is going to be applied.

§ 5.4 Bayesian model

In this section, we describe the hierarchical Bayesian model attached to the variables introduced in Section 5.1. This model follows the same strategy as in [Zhao et al., 2016] and is illustrated in Figure 5.1.

5.4.1 Priors

The likelihood corresponding to the RF image $\boldsymbol{y}$, which follows from the Gaussian nature of the noise, is derived from (5.1) and expressed below.

$$\mathrm{p}(\boldsymbol{y}|\boldsymbol{x},\sigma^2) = \frac{1}{(2\pi\sigma^2)^{m/2}}\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma^2}\right)$$

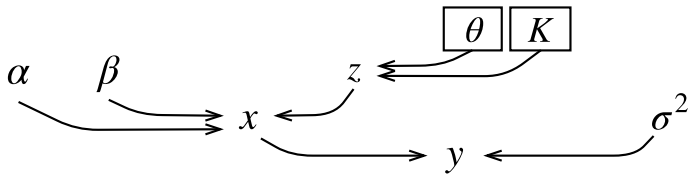


Figure 5.1: *Hierarchical Bayesian model. Parameters in boxes are fixed in advance.*

The TRF is a mixture of $\mathcal{GGD}$s which, under the assumption that the pixel values are independent given the segmentation labels $\boldsymbol{z}$, leads to the following prior

$$\mathrm{p}(\boldsymbol{x}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{z}) = \prod_{i=1}^{m} \frac{1}{2\beta_{z_i}^{1/\alpha_{z_i}}\Gamma(1+1/\alpha_{z_i})} \exp\left(-\frac{|x_i|^{\alpha_{z_i}}}{\beta_{z_i}}\right), \tag{5.7}$$

where Γ is the Gamma function. For every set $\mathcal{C}$, let $\imath_{\mathcal{C}}$ denote the indicator function of this set. Uninformative Jeffreys priors are assigned to the noise variance σ^2 and scale parameters $\boldsymbol{\beta}$, while the shape parameters $\boldsymbol{\alpha}$ are assumed to be uniformly distributed between 0 and 3. These assumptions lead to the following priors for these variables,

$$\mathrm{p}(\sigma^2) = \frac{1}{\sigma^2}\imath_{\mathbb{R}_+}(\sigma^2),$$

$$\mathrm{p}(\boldsymbol{\alpha}) = \prod_{k=1}^{K} \frac{1}{3}\imath_{[0,3]}(\alpha_k), \quad \text{and} \quad \mathrm{p}(\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{1}{\beta_k}\imath_{\mathbb{R}_+}(\beta_k).$$

The segmentation labels $\boldsymbol{z}$ are modeled by a Potts Markov random field with prior

$$\mathrm{p}(\boldsymbol{z}) = \frac{1}{C(\theta)} \exp\left(\sum_{i=1}^{m} \sum_{j\in\mathcal{V}(i)} \theta\delta(z_i - z_j)\right), \tag{5.8}$$

with δ the Kronecker function, $C(\theta) \in \mathbb{R}_+^*$ a normalizing constant, $\theta \in \mathbb{R}_+^*$ a granularity coefficient, and $\mathcal{V}(i)$ the set of four closest neighbours of the ith pixel.

The different variables are sampled according to their conditional distributions, which are provided in the next section.

5.4.2 Conditional distributions

The conditional distribution of the noise variance is derived from the Bayes theorem and can be expressed as follows,

$$\mathrm{p}(\sigma^2|\boldsymbol{y},\boldsymbol{x}) \propto \mathcal{IG}\left(\frac{m}{2}, \frac{\|\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}\|^2}{2}\right), \tag{5.9}$$

where $\mathcal{IG}$ denotes the inverse gamma distribution. Assuming that the different regions have independent shape and scale parameters, for every region $k \in \{1, \dots, K\}$, we obtain

$$\mathrm{p}(\alpha_k|\boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{z}) \propto \prod_{i \in \mathcal{I}_k} \frac{\imath_{[0,3]}(\alpha_k)}{2\beta_k^{1/\alpha_k}\Gamma\left(1 + 1/\alpha_k\right)} \exp\left(-\frac{|x_i|^{\alpha_k}}{\beta_k}\right), \tag{5.10}$$

and

$$\mathrm{p}(\beta_k|\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{z}) \propto \mathcal{IG}\left(\frac{m_k}{\alpha_k}, \sum_{i \in \mathcal{I}_k} |x_i|^{\alpha_k}\right), \tag{5.11}$$

with $\mathcal{I}_k = \{i \in \{1, \dots, m\} \mid z_i = k\}$ and m_k the number of elements in $\mathcal{I}_k$. For every pixel $i \in \{1, \dots, m\}$ and every region $k \in \{1, \dots, K\}$, the Bayes rule applied to the segmentation labels leads to the following probability distribution,

$$\mathrm{p}(z_i = k|\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}_{\mathcal{V}(i)}) \propto \frac{\exp\left(\sum_{j \in \mathcal{V}(i)} \theta\delta(z_j - k) - \frac{|x_i|^{\alpha_k}}{\beta_k}\right)}{2\beta_k^{1/\alpha_k}\Gamma(1 + 1/\alpha_k)}, \tag{5.12}$$

where $\boldsymbol{z}_{\mathcal{V}(i)}$ denotes the vector made of the labels in the neighborhood of the ith pixel. Finally, the conditional distribution of the TRF is equal to

$$\pi(\boldsymbol{x}) = \mathrm{p}(\boldsymbol{x}|\boldsymbol{y}, \sigma^2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}) \propto \exp\left(-f(\boldsymbol{x}) - g(\boldsymbol{x})\right), \tag{5.13}$$

where, for every $\boldsymbol{x} \in \mathbb{R}^m$,

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \frac{|x_i|^{\alpha_{z_i}}}{\beta_{z_i}} \quad \text{and} \quad g(\boldsymbol{x}) = \frac{\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2}{2\sigma^2}. \tag{5.14}$$

§ 5.5 Hybrid Gibbs sampler

We can now describe the proposed hybrid Gibbs sampler used to perform joint deconvolution and segmentation of ultrasound images. A special attention is devoted to the sampling scheme of the TRF, which is our main contribution.

5.5.1 Sampling from the conditional distributions

Samples for the noise variance σ^2 and scale parameters $\boldsymbol{\beta}$ can easily be drawn from their respective conditional distribution (5.9) and (5.11). Regarding the shape parameters $\boldsymbol{\alpha}$, we use the Metropolis-Hastings random walk Algorithm 1, presented in Section 2.1.3.2, to generate samples according to distribution (5.10). For the segmentation labels $\boldsymbol{z}$, we first suitably

normalize the distributions given by (5.12). The resulting normalized conditional probabilities are given below

$$(\forall i \in \{1,\dots,m\})(\forall k \in \{1,\dots,K\}) \quad \tilde{p}_{i,k} = \frac{p(z_i = k|\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{z}_{\mathcal{V}(i)})}{\sum_{l=1}^{K} p(z_i = l|\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{z}_{\mathcal{V}(i)})}. \tag{5.15}$$

Hence, for every pixel $i \in \{1,\dots,m\}$, label z_i is drawn from $\{1,\dots,K\}$ using the probabilities $(\tilde{p}_{i,k})_{1\leq k\leq K}$ defined above. Finally, we propose to apply our PP-ULA scheme (5.6) to the conditional distribution (5.13) in order to generate samples for the TRF $\boldsymbol{x}$. This leads to the following scheme,

$$(\forall t \in \mathbb{N}) \quad \boldsymbol{x}^{(t+1)} = \operatorname{prox}_{\gamma f}^{\boldsymbol{M}}(\tilde{\boldsymbol{x}}^t) + \sqrt{2\gamma}\boldsymbol{M}^{-\frac{1}{2}}\boldsymbol{\omega}^{(t+1)}, \tag{5.16}$$

where f is defined as in (5.14), $\boldsymbol{M} \in \mathfrak{S}_m^+$ is a preconditioning matrix used for acceleration, $\boldsymbol{\omega}^{(t+1)} \sim \mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$, and

$$\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{x}^{(t)} - \frac{\gamma}{\sigma^2}\boldsymbol{M}^{-1}\boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{x}^{(t)} - \boldsymbol{y}). \tag{5.17}$$

It can be noted that, in Proposition 5.3.1, f is assumed to be convex, which is not necessarily satisfied in our case. However, for simplicity, we take the discrete scheme (5.16) even in the nonconvex case. Since the proposed sampling strategy is unadjusted, (5.16) is not followed by an acceptance test. The bias with respect to π increases with γ, as the speed of convergence of the algorithm. A compromise must be found when setting γ.

As described in the next section, when there exist shape parameters that are below 1, we use the MM principle [Schifano et al., 2010] introduced in Section 2.2.4 to replace the nonconvex minimization problem involved in the computation of $\operatorname{prox}_{\gamma f}^{\boldsymbol{M}}$ with a sequence of convex surrogate problems.

5.5.2 MM principle for computing the proximity operator

The function f defined in (5.14) can be written as the sum of a convex term and a nonconvex term. We propose to approximate the nonconvex part using the MM principle.

Let $\sigma \in \mathbb{R}_+^*$, $\boldsymbol{\alpha} \in [0,3]^K$, $\boldsymbol{\beta} \in (\mathbb{R}_+^*)^K$ and $\boldsymbol{z} \in \{1,\dots,K\}^m$. We define the sets $\mathcal{I}_{<1} = \{i \in \{1,\dots,m\} \mid \alpha_{z_i} < 1\}$ and $\mathcal{I}_{\geq 1} = \{1,\dots,m\} \setminus \mathcal{I}_{<1}$. Let $\mathcal{J} \subset \mathcal{I}_{<1}$. We define $h_{\mathcal{J}}$ at every $(\boldsymbol{u},\boldsymbol{v}) \in \mathbb{R}^m \times (\mathbb{R}_+^*)^m$ by

$$h_{\mathcal{J}}(\boldsymbol{u},\boldsymbol{v}) = \sum_{i\in\mathcal{I}_{\geq 1}} \frac{|u_i|^{\alpha_{z_i}}}{\beta_{z_i}} + \sum_{j\in\mathcal{J}} \frac{(1-\alpha_{z_j})v_j^{\alpha_{z_j}} + \alpha_{z_j}v_j^{\alpha_{z_j}-1}|u_j|}{\beta_{z_j}}.$$

From concavity, we deduce that, for every $\boldsymbol{v} \in (\mathbb{R}_+^*)^m$ and $\boldsymbol{u} \in \mathbb{R}^m$ such that $\mathcal{J} \subset \{i \in \mathcal{I}_{<1} \mid |u_i| > 0\}$, the following majoration property holds

$$h_{\mathcal{J}}(\boldsymbol{u},\boldsymbol{v}) \geq \sum_{i\in\mathcal{I}_{\geq 1}\cup\mathcal{J}} \frac{|u_i|^{\alpha_{z_i}}}{\beta_{z_i}} = h_{\mathcal{J}}(\boldsymbol{u},(|u_i|)_{1\leq i\leq m}).$$

Since $h_{\mathcal{J}}(\cdot, \boldsymbol{v})$ is convex and separable, its proximity operator in the Euclidean metric is straightforward to compute. More precisely, for every $i \in \mathcal{I}_{\geq 1}$, $\eta \in \mathbb{R}_+^*$ and $s \in \mathbb{R}$, $\mathrm{prox}_{\eta^{-1}|\cdot|^{\alpha z_i}}(s)$ has either a closed form, given in Example 2.2.14, or can be found using a bisection search in $[0, |s|]$.

The dual forward-backward Algorithm 5 is then combined with Remark 2.2.8 in order to compute the proximity operator of the convex function $h_{\mathcal{J}}(\cdot, \boldsymbol{v})$ in any metric. This leads to Algorithm 15 below, which generates a sequence $(\boldsymbol{s}^{(p)})_{p\in\mathbb{N}}$ converging to $\mathrm{prox}^{\boldsymbol{M}}_{h_{\mathcal{J}}(\cdot,\boldsymbol{v})}(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{M} \in \mathfrak{S}_m^+$.

Algorithm 15: DFB algorithm to compute $\mathrm{prox}^{\boldsymbol{M}}_{h_{\mathcal{J}}(\cdot,\boldsymbol{v})}(\boldsymbol{x})$

Set $\rho = 1/\left|\left|\left|\boldsymbol{M}^{-1}\right|\right|\right|$, $\epsilon \in]0, \min\{1, \rho\}[$, $\eta \in [\epsilon, 2\rho - \epsilon]$;
Initialization: dual variable $\boldsymbol{w}^{(1)} \in \mathbb{R}^m$;
for $p = 1, \ldots$ **do**
  $\boldsymbol{s}^{(p)} = \boldsymbol{x} - \boldsymbol{M}^{-1}\boldsymbol{w}^{(p)}$;
  $\boldsymbol{w}^{(p+1)} = \boldsymbol{w}^{(p)} + \eta\boldsymbol{s}^{(p)} - \eta\mathrm{prox}_{\eta^{-1}h_{\mathcal{J}}(\cdot,\boldsymbol{v})}(\eta^{-1}\boldsymbol{w}^{(p)} + \boldsymbol{s}^{(p)})$
end

Finally, Algorithm 16 generates a sequence $(\boldsymbol{u}^{(q)})_{q\in\mathbb{N}}$ estimating $\mathrm{prox}^{\boldsymbol{M}}_{\gamma f}(\tilde{\boldsymbol{x}}^{(t)})$.

Algorithm 16: MM principle to compute $\mathrm{prox}^{\boldsymbol{M}}_{\gamma f}$.

Initialize $\boldsymbol{u}^{(1)} \in \mathbb{R}^m$;
for $q = 1, \ldots$ **do**
  $\mathcal{J}^{(q)} = \{i \in \mathcal{I}_{<1} \mid |u_i^{(q)}| > 0\}$;
  $\boldsymbol{v}^{(q)} = (|u_i^{(q)}|)_{1\leq i\leq m}$;
  $\boldsymbol{u}^{(q+1)} = \mathrm{prox}^{\boldsymbol{M}}_{\gamma h_{\mathcal{J}^{(q)}}(\cdot,\boldsymbol{v}^{(q)})}(\tilde{\boldsymbol{x}}^{(t)})$ (using Algorithm 15)
end

5.5.3 Proposed sampler

The resulting hybrid Gibbs sampler is summarized in Algorithm 17. It is evaluated on six different experiments in the next section.

Algorithm 17: Hybrid Gibbs sampler

1 Sample the noise variance σ^2 according to (5.9);

2 Sample the shape parameter $\boldsymbol{\alpha}$ using Algorithm 1 applied to (5.10);

3 Sample the scale parameter $\boldsymbol{\beta}$ according to (5.11);

4 Sample the hidden label field $\boldsymbol{z}$ using (5.15) ;

5 Sample the TRF $\boldsymbol{x}$ using PP-ULA (5.16)-(5.17).

§ 5.6 Numerical experiments

5.6.1 Experimental settings

Simu1 and Simu2 refer to simulated images with two and three regions, respectively. Kidney denotes the tissue-mimicking phantom produced from 10^6 scatterers uniformly distributed over a digital image of human kidney tissue provided with the Field II ultrasound simulator [Jensen, 2004]. The amplitude of each scatterer is produced using a zero-mean Gaussian distribution, whose variance is linked to the amplitude of the corresponding point on the digital image. The PSF for the aforementioned simulations is obtained with Field II and corresponds to a 3.5 MHz linear probe. The number of regions K is set to 2 for Simu1, and it is set to 3 for Simu2 and Kidney.

We also perform tests on three real ultrasound images. Thyroid denotes a real RF image of thyroidal flux obtained *in vivo* with a 7.8 MHz probe. The unknown PSF is identified using the RF image of a wire cross-section which was acquired with the same probe. Since the diameter of the wire is of the order of a few µm, its cross-section can almost be viewed as a point. Thus, its RF image provides a good approximation of the PSF. Finally, Bladder and KidneyReal refer to the RF images of a mouse bladder and mouse kidney, respectively. Both images were obtained *in vivo* with a 20 MHz probe. The PSF for these two real images is estimated using the same method as for Thyroid. The number of regions K is set to 2 for KidneyReal, and it is set to 3 for Thyroid and Bladder.

Test settings are summarized Table 5.1, while the simulated and real RF images used in these numerical experiments are shown in Figure 5.2.

The TRF is initialized using a pre-deconvolved image obtained with a Wiener filter, while the segmentation is initialized by applying a 7×7 median filter, followed by the Otsu method [Otsu, 1979], to the B-mode of the initial TRF. Shape and scale parameters are randomly selected in $[0.5, 1.5]$, and $[1, 200]$, respectively. The granularity parameter θ for the Potts model (5.8) is adjusted to ensure that the percentage of isolated points in the segmentation, obtained with a 3×3 median filter, is close to 0.05, 0.1, 0.8, 0.08, 0.08 and 0.08 for Simu1, Simu2, Kidney, Thyroid, Bladder and KidneyReal, respectively.

Experiment	Size	Data type	Ground-truth		
			$\boldsymbol{x}$	$(\boldsymbol{\alpha}, \boldsymbol{\beta})$	$\boldsymbol{z}$
Simu1	256×256	Simulated	✓	✓	✓
Simu2	256×256	Simulated	✓	✓	✓
Kidney	294×354	Tissue-mimicking	✓	-	-
Thyroid	870×140	Real *in vivo*	-	-	-
Bladder	370×256	Real *in vivo*	-	-	-
KidneyReal	350×200	Real *in vivo*	-	-	-

Table 5.1: *Test settings: size of test images, data type, and availability of the ground-truth for the TRF, $\mathcal{GGD}$ parameters and the segmentation.*

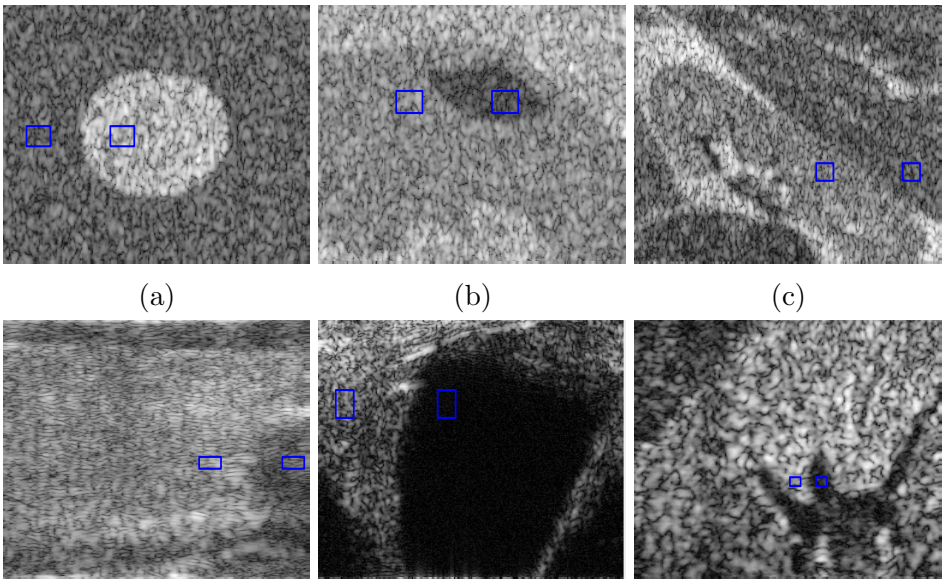


Figure 5.2: *B-mode of RF images for (a) Simu1, (b) Simu2, (c) Kidney, (d) Thyroid, (e) Bladder and (f) KidneyReal. Blue boxes indicate regions used to compute the CNR.*

5.6.2 Comparisons and evaluation metrics

All computational times are given for simulations run on Matlab 2018b on an Intel Xeon CPU E5-1650 3.20 GHz. The code for the proposed method is available online[3]. In addition to comparing Algorithm 17 with HMC [Zhao et al., 2016], the quality of the deconvolution is compared with the one

[3] https://github.com/mccorbineau/PP-ULA

obtained with a Wiener filter, where the noise level has been estimated as in [Mallat, 1999], and with the solution to the Lasso problem, where the regularization weight is set *i*) manually when the ground-truth is not available, or *ii*) using a golden-section search to maximize the peak signal-to-noise ratio (PSNR) defined as

$$\mathrm{PSNR} = 10 \log_{10} \left(m \frac{\max_{1 \leq i \leq m} (\overline{x}_i, x_i^*)^2}{\|\overline{\boldsymbol{x}} - \boldsymbol{x}^*\|^2} \right), \tag{5.18}$$

with $\overline{\boldsymbol{x}} \in \mathbb{R}^m$ the true TRF and $\boldsymbol{x}^* \in \mathbb{R}^m$ the estimated one. We also compare our results with the segmentation given by Otsu's method [Otsu, 1979] applied to the Wiener-deconvolved image, and with the SLaT method [Cai et al., 2017] applied to the Lasso-deconvolved image. The proposed method, PP-ULA, is used with $\gamma = 0.09$ and $\boldsymbol{M}$ an approximation of the Hessian of the differentiable term in (5.13) [Becker and Fadili, 2012], $\boldsymbol{M} = (\boldsymbol{H}^\top \boldsymbol{H} + \lambda \boldsymbol{I}_m)/\sigma^2$, with $\lambda = 0.1$ so that the inverse of $\boldsymbol{M}$ is well-defined.

In order to evaluate the deconvolution results, we compute the PSNR and the structural similarity measure (SSIM) [Wang et al., 2004] between the restored TRF and its ground-truth. The contrast in the restored TRF is assessed according to the contrast-to-noise ratio (CNR) [Krishnan et al., 1997] between two windows from different regions of the B-mode TRF images. The CNR is defined as follows,

$$\mathrm{CNR} = |\mu_1 - \mu_2| / (\nu_1 + \nu_2)^{1/2},$$

where (μ_1, μ_2) and (ν_1, ν_2) are, respectively, the means and variances of the two windows. The segmentation is evaluated according to the percentage of correctly predicted labels, which is referred to as overall accuracy (OA). The minimum mean square error (MMSE) estimators of all parameters in HMC and PP-ULA are computed after the burn-in regime. Moreover, to evaluate the mixing property of the Markov chain after convergence, we compute the mean square jump (MSJ) per second, which is the ratio of the MSJ to the time per iteration. The MSJ is obtained using T samples of the TRF $(\boldsymbol{x}^{(t_0+1)}, \ldots, \boldsymbol{x}^{(t_0+T)})$ generated after the burn-in period, i.e.

$$\mathrm{MSJ} = \left(\frac{1}{T-1} \sum_{t=1}^{T-1} \left\| \boldsymbol{x}^{(t_0+t)} - \boldsymbol{x}^{(t_0+t+1)} \right\|^2 \right)^{1/2}.$$

First, we present the results obtained on the proposed simulated data.

5.6.3 Results on simulated data

The convergence speed of Algorithm 17 is empirically observed for Simu1 and Simu2, as illustrated in Figure 5.3, where we also display the results of the non-preconditioned P-ULA, for which $\boldsymbol{M} = \boldsymbol{I}_m$ and $\gamma = 1.99\sigma^2 / |||\boldsymbol{H}|||^2$.

		Iterations		Time		Mixing
		Burn-in	Total	Duration	PP-ULA speed gain	MSJ (per sec)
Simu1	P-ULA	70000	140000	2 h 27 min	12.2	665
	HMC	4000	8000	1 h 08 min	5.7	173
	PP-ULA	4000	8000	12 min	1	970
Simu2	P-ULA	70000	140000	3 h 06 min	4.8	590
	HMC	10000	20000	4 h 14 min	6.6	22
	PP-ULA	10000	20000	39 min	1	793

Table 5.2: *Number of iterations, computational time and MSJ per sec for experiments Simu1 and Simu2.*

	Simu1					Simu2						
	σ^2	α_1	β_1	α_2	β_2	σ^2	α_1	β_1	α_2	β_2	α_3	β_3
True	0.013	1.5	1.0	0.60	1.0	33	1.5	100	1.0	50	0.50	4.0
P-ULA	0.041	2.0	0.5	0.59	1.0	122	2.0	330	2.0	3186	0.48	3.4
HMC	0.013	1.8	1.2	0.61	1.0	34	1.4	66	1.1	111	0.54	5.2
PP-ULA	0.013	1.4	0.9	0.62	1.1	35	2.3	2676	1.2	122	0.55	5.8

Table 5.3: *MMSE Estimates of the noise variance and $\mathcal{GGD}$ parameters.*

Comparing P-ULA and PP-ULA on these simulated data allows us to study the effect of adding a preconditioner in the proposed sampling scheme. As reported in Table 5.2, P-ULA needs more iterations and more time to converge than PP-ULA: the proposed method is 12.2 and 4.8 times faster than P-ULA on Simu1 and Simu2, respectively. In addition, from Table 5.3 and Figure 5.4, we deduce that P-ULA is more biased than PP-ULA, which samples correctly the target distributions. Finally, as one can see in Figure 5.3 and Table 5.4, P-ULA leads to lower PSNR, SSIM and OA values than PP-ULA. These results clearly emphasize the benefits of preconditioning in this example.

From Table 5.2, PP-ULA is 5.7 and 6.6 times faster than HMC on Simu1 and Simu2 and has better mixing properties, as shown by the MSJ per second. Visual results from Figures 5.5 and 5.6, and CNR values in Table 5.4 show that the contrast obtained with PP-ULA is better than with competitors on Simu2, and is second best after P-ULA on Simu1. However, it should be noted that the PSNR and SSIM obtained on Simu1 with P-ULA are much lower than with the other methods. In addition, the PSNR and

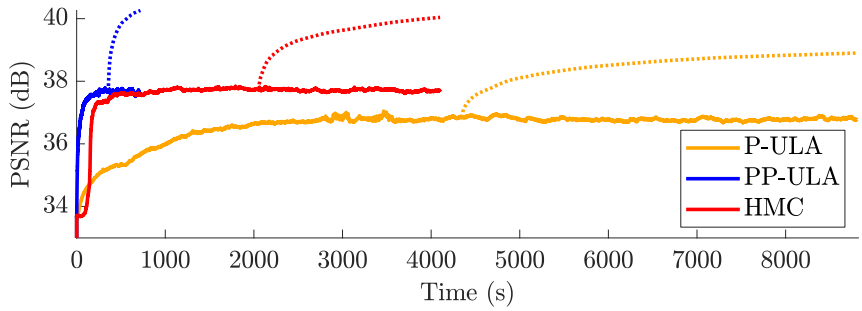


Figure 5.3: *PSNR along time for Simu1. Dotted lines indicate the PSNR of the MMSE estimator of the TRF after the burn-in regime.*

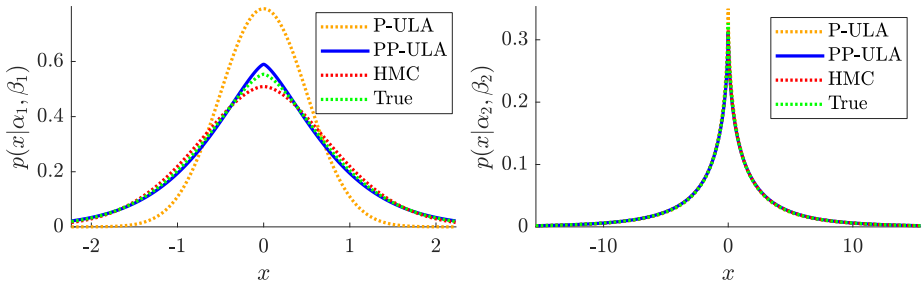
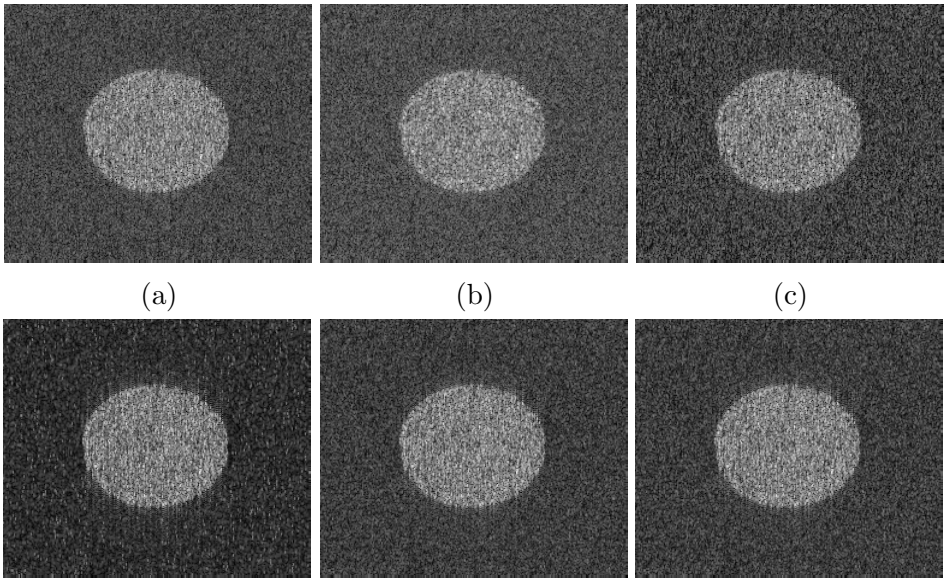


Figure 5.4: *Simu1, $\mathcal{GGD}$ distributions (5.7) of regions 1 (left) and 2 (right).*

	Simu1				Simu2			
	PSNR	SSIM	CNR	OA	PSNR	SSIM	CNR	OA
Wiener - Otsu	37.1	0.57	1.26	99.5	35.4	0.63	0.97	96.0
Lasso - SLaT	39.2	0.60	1.15	99.6	37.8	0.70	0.99	98.3
P-ULA	38.9	0.45	**1.82**	98.7	37.1	0.57	1.59	94.9
HMC	40.0	**0.62**	1.47	**99.7**	36.4	0.64	1.59	98.5
PP-ULA	**40.3**	**0.62**	1.51	**99.7**	**38.6**	**0.71**	**1.64**	**98.7**

Table 5.4: *PSNR, SSIM, CNR and segmentation OA for simulated data.*

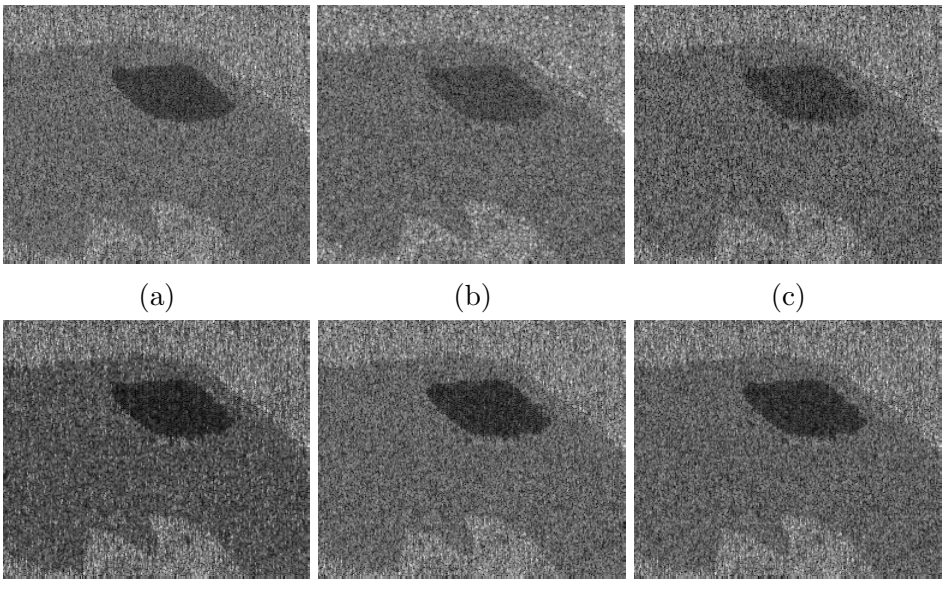
SSIM values from Table 5.4 obtained with PP-ULA are equivalent or higher than all competitors for these two experiments. Visual segmentation results are shown in Figures 5.7 and 5.8, and OA values can be found in Table 5.4. For these simulated images, more pixels are correctly labeled with PP-ULA than with competitors.



(a) (b) (c)

(d) (e) (f)

Figure 5.5: *B-mode visualization of the TRF for Simu1: (a) ground-truth, (b) Wiener, (c) Lasso, (d) P-ULA, (e) HMC, (f) PP-ULA.*



(a) (b) (c)

(d) (e) (f)

Figure 5.6: *B-mode visualization of the TRF for Simu2: (a) ground-truth, (b) Wiener, (c) Lasso, (d) P-ULA, (e) HMC, (f) PP-ULA.*

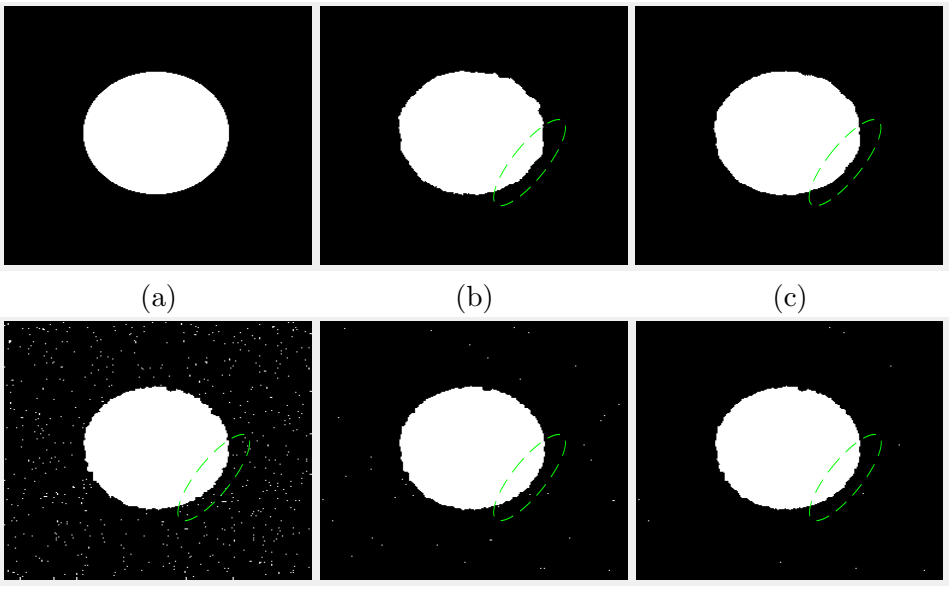


Figure 5.7: *Segmentation results for Simu1: (a) ground-truth, (b) Otsu, (c) SLaT, (d) P-ULA, (e) HMC, (f) PP-ULA. Main differences are circled in green.*

		Iterations		Time		Mixing
		Burn-in	Total	Duration	PP-ULA speed gain	MSJ (per sec)
Kidney	HMC	7000	14000	4 h 23 min	6.3	167
	PP-ULA	7000	14000	42 min	1	657
Thyroid	HMC	3000	6000	2 h 09 min	3.7	175
	PP-ULA	3000	6000	35 min	1	950
Bladder	HMC	5000	10000	2 h 45 min	5.2	13
	PP-ULA	5000	10000	32 min	1	1396
KidneyReal	HMC	5000	10000	1 h 49 min	5.8	11
	PP-ULA	5000	10000	19 min	1	1361

Table 5.5: *Number of iterations, computational time and MSJ per sec for experiments on the tissue-mimicking phantom and on real data.*

5.6.4 Results on a tissue-mimicking phantom and on real data

The convergence of Algorithm 17 is also empirically observed for the experiments on the tissue-mimicking phantom and on real data, i.e. Kidney,

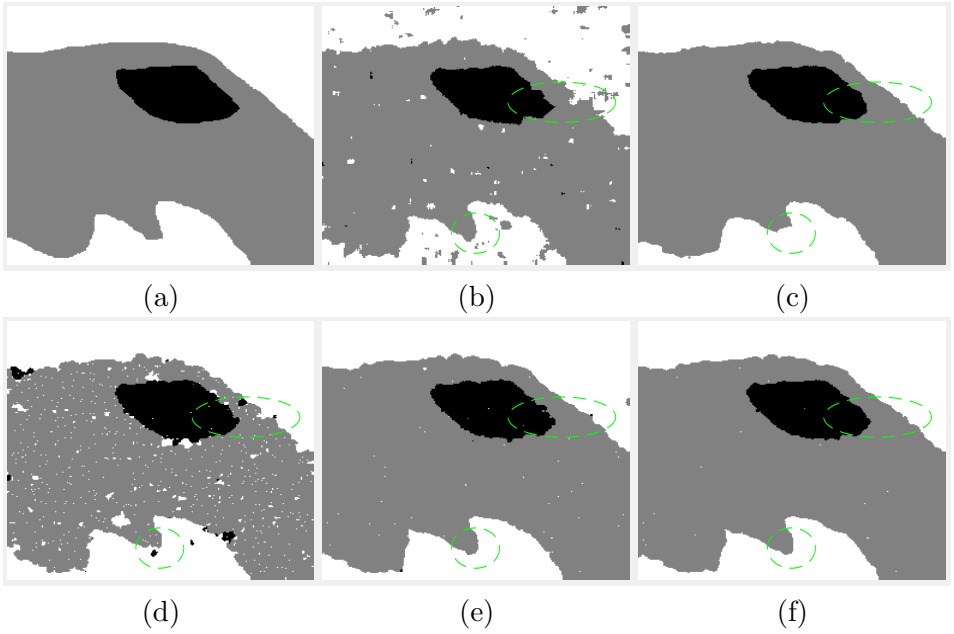


Figure 5.8: *Segmentation results for Simu2: (a) ground-truth, (b) Otsu, (c) SLaT, (d) P-ULA, (e) HMC, (f) PP-ULA. Main differences are circled in green.*

Thyroid, Bladder and KidneyReal. As mentioned in Table 5.5, the proposed method leads to a significant acceleration since it is between 3.7 and 6.3 times faster than HMC on these experiments. Visual results from Figures 5.9 and 5.10, and CNR values in Table 5.6 show that the contrast obtained with PP-ULA is better than with competitors on all these test images. In addition, the PSNR and SSIM values from Table 5.6 obtained with PP-ULA on the Kidney experiment are equivalent or higher than all competitors.

	Kidney			Thyroid	Bladder	KidneyReal
	PSNR	SSIM	CNR	CNR	CNR	CNR
Wiener	27.6	0.58	0.66	0.56	1.66	1.61
Lasso	28.5	0.59	0.67	0.99	1.76	1.76
HMC	**29.5**	**0.62**	1.10	1.52	2.23	1.88
PP-ULA	29.3	**0.62**	**1.14**	**1.56**	**2.48**	**1.93**

Table 5.6: *PSNR, SSIM and CNR results.*

Although the ground-truth of the segmentation is not available for these experiments, one can see from the visual segmentation results shown in

Figure 5.11, that the segmentation based on the Potts model (PP-ULA and HMC) gives more homogeneous areas than Otsu, and recovers more details than SLaT.

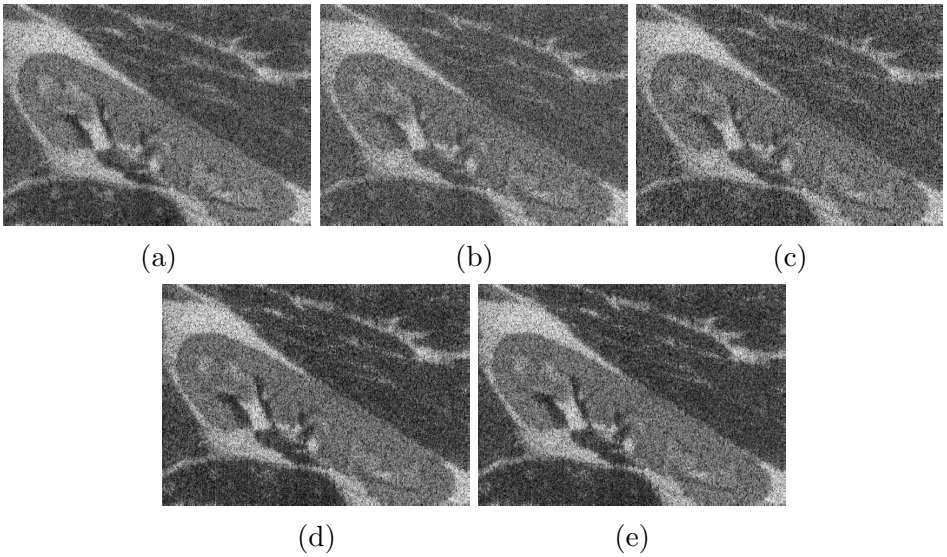
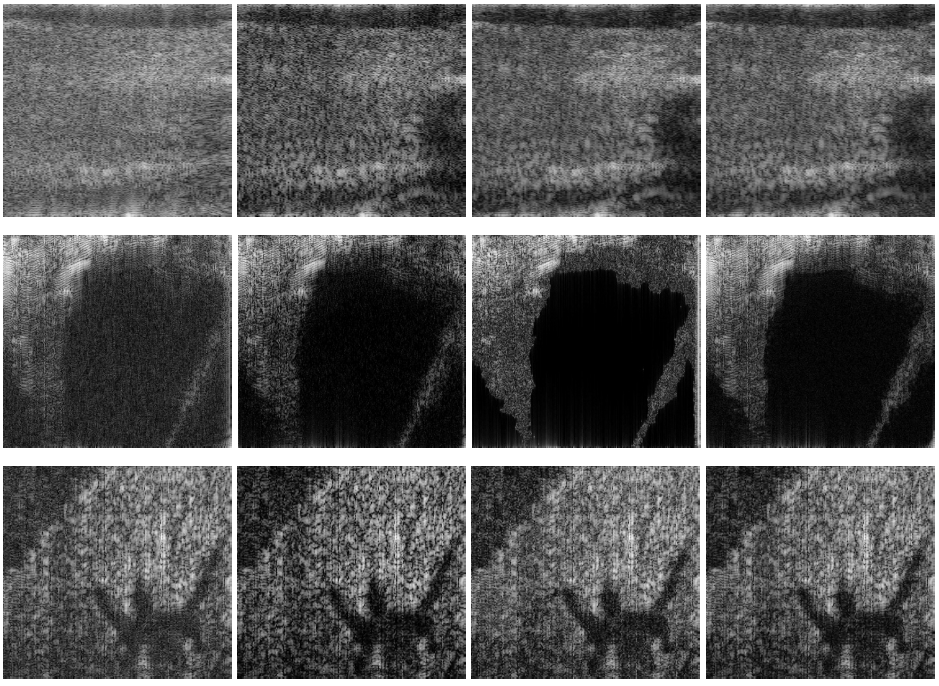


Figure 5.9: *B-mode visualization of the TRF results for Kidney. (a) ground-truth, (b) Wiener, (c) Lasso, (d) HMC, (e) PP-ULA.*

§ 5.7 Summary

In this chapter, we have investigated a new method based on a preconditioned proximal unadjusted Langevin algorithm for the joint restoration and segmentation of ultrasound images, which showed faster convergence than an existing Hamiltonian Monte Carlo algorithm. Hence, the proposed method has the potential to speed-up the approach proposed in [Pereyra et al., 2012] for the segmentation of ultrasound images. Another direction for future work is to extend this framework to a spatially variant, possibly unknown, PSF.



(a) (b) (c) (d)

Figure 5.10: *B-mode visualization of the TRF results for real data: (a) Wiener, (b) Lasso, (c) HMC, (d) PP-ULA. From top to bottom: Thyroid, Bladder and KidneyReal.*

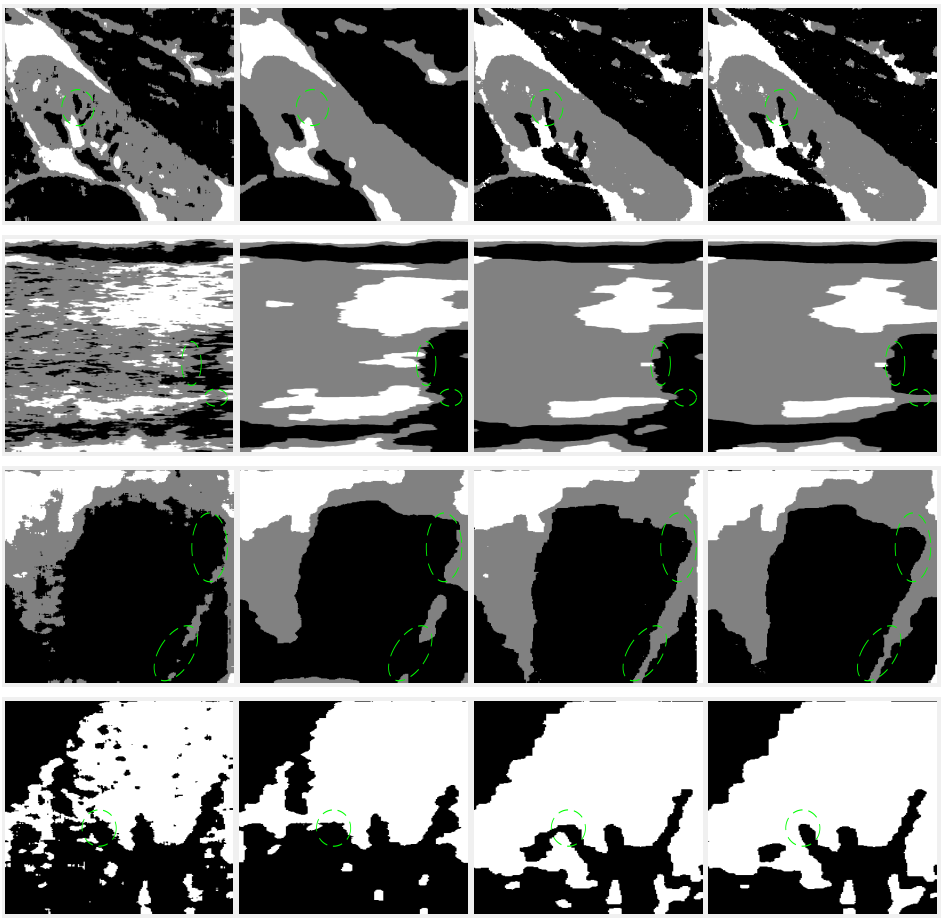


Figure 5.11: *Segmentation: (a) Otsu, (b) SLaT, (c) HMC, (d) PP-ULA. Top to bottom: Kidney, Thyroid, Bladder, KidneyReal. Main differences are circled in green.*

- Chapter 6 -

Conclusion

§ 6.1 Summary

During this thesis, our main axes of research aimed at studying the use of the proximity operator in different contexts, and at building bridges between distinct methods through optimization. Accordingly, we have investigated three new methods for solving challenging inverse problems.

First, we have focused on the traditional variational strategy, which consists in finding a solution to an inverse problem by minimizing an objective function. Interior point methods are a specific class of variational methods that are well suited for constrained optimization problems. Despite their impressive performance in linear and quadratic programming, they progressively lost popularity to give rise to more flexible techniques, such as proximal algorithms.

In Chapter 3, we have combined the logarithmic barrier method with the proximal variable metric forward-backward algorithm, leading to the proposed PIPA algorithm. In the case of composite objective functions, PIPA benefits from a splitting strategy. In addition, it includes variable metrics, which can be used to boost its convergence as proven in our numerical experiments. Last but not least, the logarithmic barrier in PIPA ensures that every iterate satisfies strictly the constraints, which might be critical if, for instance, these constraints model a safety margin or if they have a physical meaning.

We have then carried out a mathematical analysis of the proposed algorithm. The convergence of PIPA was obtained under mild boundedness conditions on the variable metrics, and we have shown that, under some assumptions, the inner loop in PIPA converges linearly. Finally, we have evaluated the performance of PIPA on two applications in image processing. The first example concerns hyperspectral unmixing and the second one

addresses joint geometry-texture decomposition and reconstruction of CT data. In both experiments, we have formulated a constrained optimization problem, with a nondifferentiable term in the objective function, for solving the corresponding inverse problem. PIPA compares favorably with widely-used state-of-the-art algorithms on these two applications.

Second, we have considered the framework offered by recent deep learning methods for solving inverse problems. Neural networks have reached outstanding performances in applications related to computer vision. They may however suffer from a lack of interpretability and mathematical guarantees.

In Chapter 4, we have applied deep-unfolding to a proximal interior point algorithm. This algorithm has been unfolded over a fixed number of iterations identified with the network layers. As opposed to other unfolded methods, we have maintained the link between the original algorithm and the resulting architecture by keeping the proximal update explicit, so that only hyperparameters are inferred by the network.

In order to train the proposed network, referred to as iRestNet, using gradient descent and backpropagation, we have derived explicit expressions for the proximity operator of the barrier and for its first derivatives for three common types of constraints. It is worth noting that the logarithmic barrier allows to incorporate prior knowledge or desired properties about the solution within iRestNet. Moreover, when the optimization problem is quadratic, we have derived sufficient conditions under which iRestNet is robust with regards to an input perturbation. Such stability property is very important to promote the use of deep neural networks in fields involving high responsibility, such as medical imaging.

Lastly, we have demonstrated the good performance of iRestNet on image deblurring for several types of blurring kernels and noise levels. It is worth noting that iRestNet compares favorably in terms of image restoration with the classical variational approach and state-of-the-art neural network approaches for image deblurring, including architectures derived from other types of unfolded algorithms.

Third, we have combined proximal tools with a Bayesian approach. The Bayesian framework provides useful stochastic simulation methods, that can be used as alternatives to variational methods which may require setting hyperparameters. In addition, Bayesian methods can help derive useful estimators, other than the MAP, such as the MMSE. However, Bayesian techniques can suffer from high computational costs.

In Chapter 5, we have introduced a new accelerated proximal unadjusted Langevin algorithm called PP-ULA. The preconditioning matrices and splitting scheme used in PP-ULA constitute two original features of this algorithm. Since PP-ULA is an unadjusted sampling method, it is not followed by

an acceptance test. Then, we have considered a hybrid Gibbs sampler which has been derived for a specific application, the deconvolution and segmentation of ultrasound images. We have incorporated PP-ULA into this Gibbs sampler so as to accelerate the method. Finally, we have conducted tests on six different images, half of which are synthetic images, and the remaining are real *in vivo* ultrasound data. Using the proposed PP-ULA has led to a significant reduction of the computational time compared to an Hamiltonian Monte Carlo algorithm known for its good performance. In addition, the proposed method also produced better deconvolution and segmentation results than several variational approaches used as comparisons.

In the next section, we propose several extensions of the aforementioned methods that could be investigated for future works.

§ 6.2 Perspectives

The proximal interior point algorithm studied in Chapter 3 could be further improved as suggested hereafter.

Study the convergence of PIPA for nonconvex functions One interesting lead to extend the scope of application of PIPA would be to conduct its convergence analysis in a nonconvex setting. This could be done for instance by better relying on the KL property, as in the following works [Attouch and Bolte, 2009; Attouch et al., 2010; Chouzenoux et al., 2014b].

Consider inexact proximity operators in PIPA In our numerical experiments from Sections 3.6 and 3.7, we use an iterative scheme to compute the proximity operator in the variable metric. Despite this inexact computation, we still observe the practical convergence of PIPA in both applications. Therefore, another improvement in the analysis of this algorithm would be to prove mathematically its convergence assuming that the proximity operator is computed in an inexact manner. Inexact proximity operators have been addressed for example in [Salzo and Villa, 2012], where the authors consider several types of errors. It is worth noting that, depending on the error type, the sequence of errors does not necessarily need to be summable for convergence to hold.

Derive accelerated versions of PIPA Based on the seminal work [Nesterov, 1983], accelerated versions of proximal algorithms have been recently proposed, for example in [Calatroni and Chambolle, 2019]. It would be interesting to integrate these acceleration techniques into

PIPA and to perform numerical experiments to test if the resulting algorithm performs better than the simple version of PIPA.

Conduct experiments with nonlinear constraints It should be noted that the convergence results presented in Section 3.4 do not require the constraints to be linear. However, in the numerical experiments presented in Sections 3.6 and 3.7, constraints are taken linear. Hence, it would be insightful to use PIPA for an application that includes nonlinear constraints, like quadratic constraints for instance. This new setting could influence the choice of the variable metrics since it would modify the Hessian of the logarithmic barrier.

Building on the neural network architecture presented in Chapter 4, we propose the following research directions related to deep learning methods.

Extend the stability result of iRestNet The result obtained in Section 4.5 regarding the robustness of iRestNet holds under the assumption that the objective function is quadratic. Providing mathematical guarantees regarding the behavior of a neural network under an input perturbation is a prerequisite in many real-world applications. Therefore, we could consider studying the robustness of iRestNet in a more general setting so as to widen the scope of this method. In addition, this result could be improved by considering the hidden and post-processing layers in the study.

Improve the architecture of the hidden layers The proposed neural network makes use of very simple hidden layers to predict the hyperparameters involved in the unfolded algorithm, namely the stepsize, the barrier parameter, and the regularization coefficient. Using more sophisticated and deeper hidden layers could improve the restoration results. For instance, we could make the stepsize depend on the input image, as it is done for the barrier parameter. Moreover, the regularization parameter could depend on additional features in the image such as the mean, the variance, or second order spatial derivatives. We could also treat the parameter δ used to smooth the total variation as a learnable parameter.

Perform experiments with more complex models The numerical examples presented in Chapter 4 could be supplemented by additional experiments with different noise models. For instance, we could consider a Poisson noise or a Poisson-Gaussian mixture noise, which would lead to a different data fidelity function f as detailed in [Chouzenoux et al., 2015]. Furthermore, as opposed to classical variational methods, iRestNet is not limited by the number of hyperparameters. Hence, considering the local total variation used in Section 4.6, it would be interesting to introduce one regularization weight per pixel, or per

group of pixels, and to measure the influence of this modification on the quality of the deblurred image. More sophisticated regularization functions could also be used, such as nonlocal regularization functions [Peyré et al., 2008], which have demonstrated better performances than local strategies for several examples in image restoration [Kindermann et al., 2005; Li et al., 2017].

Extend iRestNet to blind deconvolution One limitation of the proposed architecture is that we need to train one network per blurring kernel. It would be interesting to include the inference of the observation operator $\boldsymbol{H}$ in iRestNet for the purposes of bringing more flexibility to the method and reducing the training time.

Finally, we suggest the following improvements and future leads related to our study of a Bayesian approach in Chapter 5.

Include the estimation of the granularity The hierarchical Bayesian model introduced in Section 5.4 includes a granularity coefficient in the segmentation Potts model. The segmentation result highy depends on the value of this parameter. In our numerical experiment we set it in a semi-automatic way that requires launching several Markov chains. The proposed method could be improved by considering the granualrity as a random variable and estimating it jointly with the other variables in the Gibbs sampler. This could however slow down the simulation.

Extend the Bayesian approach to nonblind deconvolution For real *in vivo* ultrasound data, the PSF of the probe is not known. In our experiments, we have estimated the PSF from the RF image of a thin wire. It would be interesting to include the estimation of the PSF in the proposed model, so as to improve the performance of the method on real data.

Study the convergence of PP-ULA Studying the convergence of the full hybrid Gibbs sampler is very challenging since PP-ULA is an unadjusted algorithm. Nonetheless, we could start by studying the convergence of PP-ULA alone by building on the results derived for the classical proximal unadjusted Langevin algorithm in [Pereyra, 2016].

Investigate parallelization Although PP-ULA leads to a significant reduction of the time needed to perform joint deconvolution and segmentation of US images, the computational cost remains too long for online applications. We have implemented our method using Matlab. The proposed approach would greatly benefit from an implementation in a language with better performance. In addition, we have not made use of any parallelization or GPU implementation, which could be promising research directions for improving the efficiency of PP-ULA.

List of Algorithms

List of Figures

List of Tables

Bibliography

M. Alessandrini, S. Maggio, J. Porée, L. De Marchi, N. Speciale, E. Franceschini, O. Bernard, and O. Basset. A restoration framework for ultrasonic tissue characterization. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 58(11), 2011. 111, 112

F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995. 32

M. Allain, J. Idier, and Y. Goussard. On global and local convergence of half-quadratic algorithms. *IEEE Transactions on Image Processing*, 15 (5):1130–1142, 2006. 103

F. Alvarez and A. Cabot. Asymptotic selection of viscosity equilibria of semilinear evolution equations by the introduction of a slowly vanishing term. *Discrete and Continuous Dynamical Systems*, 15(3):921, 2006. 47

B. Amos and J. Z. Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proc. of the International Conference on Machine Learning*, pages 136–145, 6–11 Aug 2017. 82

M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *Proc. of Advances in Neural Information Processing Systems*, pages 3981–3989, Barcelona, Spain, 5–10 Dec 2016. 82

P. Armand, J.-C. Gilbert, and S. Jan-Jégou. A feasible BFGS interior point algorithm for solving convex minimization problems. *SIAM Journal on Optimization*, 11(1):199–222, 2000. 41

H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009. 19, 21, 42, 135

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010. 19, 21, 135

H. Attouch, M.-O. Czarnecki, and J. Peypouquet. Coupling forward–backward with penalty schemes and parallel splitting for constrained variational inequalities. *SIAM Journal on Optimization*, 21(4):1251–1274, 2011a. 46, 47

H. Attouch, M.-O. Czarnecki, and J. Peypouquet. Prox-penalization and splitting methods for constrained variational problems. *SIAM Journal on Optimization*, 21(1):149–173, 2011b. 46

H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013. 42, 50, 59

H. Attouch, A. Cabot, and M.-O. Czarnecki. Asymptotic behavior of nonautonomous monotone and subgradient evolution equations. *Transactions of the American Mathematical Society*, 370(2):755–790, 2018. 46

J.-F. Aujol. Some first-order algorithms for total variation based image restoration. *Journal of Mathematical Imaging and Vision*, 34(3):307–327, July 2009. 11

J.-F. Aujol and T. F. Chan. Combining geometrical and textured information to perform image classification. *Journal of Visual Communication and Image Representation*, 17(5):1004–1023, 2006. 71

H. Ayasso and A. Mohammad-Djafari. Joint NDT image restoration and segmentation using Gauss–Markov–Potts prior models and variational Bayesian computation. *IEEE Transactions on Image Processing*, 19(9): 2265–2277, 2010. 112

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012. 11, 23

M. S. Bartlett. An inverse matrix adjustment arising in discriminant analysis. *The Annals of Mathematical Statistics*, 22(1):107–111, 1951. 90

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, New York, USA, 2017. doi: 10.1007/978-3-319-48311-5. 19, 22, 24, 25, 26, 52, 96, 98

S. Becker and J. Fadili. A quasi-Newton proximal splitting method. In *Proc. of Advances in Neural Information Processing Systems*, pages 2618–2626, Lake Tahoe, USA, 3–8 Dec 2012. 21, 68, 123

J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle. A ℓ_1-unified variational framework for image restoration. In *Proc. of the European Conference on Computer Vision*, pages 1–13, Prague, Czech Republic, 11–14 May 2004. Springer. 11

J.Y. Bello Cruz and T.T.A. Nghia. On the convergence of the forward–backward splitting method with linesearches. *Optimization Methods and Software*, 31(6):1209–1238, 2016. 42, 46

F. Benvenuto, A. La Camera, C. Theys, A. Ferrari, H. Lantéri, and M. Bertero. The study of an iterative method for the reconstruction of images corrupted by Poisson and Gaussian noise. *Inverse Problems*, 24(3):035016, 2008. 9

O. Bernard, J. D'hooge, and D. Friboulet. Statistics of the radio-frequency signal based on K distribution with application to echocardiography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 53(9): 1689–1694, 2006. 111

J. M. Bernardo and A. F. M. Smith. *Bayesian theory*, volume 405. John Wiley & Sons, Hoboken, USA, 2009. 12

M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8): 882–889, 2003. 71

M. Bertero and M. Piana. Inverse problems in biomedical imaging: modeling and methods of solution. In *Complex Systems in Biomedicine*, pages 1–33. Springer, Milano, Italy, 2006. 9

M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini. Image deblurring with Poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006, 2009. 9, 10

J. Besag, J. York, and A. Mollié. Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1):1–20, 1991. 12

J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, 2012. 67

D. Böhning and B. G. Lindsay. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663, 1988. 31

J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007. 19

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014. 19, 20

S. Bonettini and M. Prato. New convergence results for the scaled gradient projection method. *Inverse Problems*, 31(9):1–20, 2015a. 84

S. Bonettini and M. Prato. New convergence results for the scaled gradient projection method. *Inverse Problems*, 31(9), 2015b. 43

S. Bonettini and T. Serafini. Non-negatively constrained image deblurring with an inexact interior point method. *Journal of Computational and Applied Mathematics*, 231(1):236–248, 2009a. 41

S. Bonettini and T. Serafini. Non-negatively constrained image deblurring with an inexact interior point method. *Journal of Computational and Applied Mathematics*, 231(1):236 – 248, 2009b. 84

S. Bonettini, I. Loris, F. Porta, and M. Prato. Variable metric inexact line-search-based methods for nonsmooth optimization. *SIAM journal on optimization*, 26(2):891–921, 2016. 42, 43

J.-F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, Berlin, Germany, 2013. 36, 65

J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, Berlin, Germany, 2006. 36

D. Boublil, M. Elad, J. Shtok, and M. Zibulevsky. Spatially-adaptive reconstruction in computed tomography using neural networks. *IEEE Transactions on Medical Imaging*, 34(7):1474–1485, 2015. 16

C. A. Bouman and K. Sauer. A unified approach to statistical tomography using coordinate descent optimization. *IEEE Transactions on Image Processing*, 5(3):480–492, 1996. 11

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004. 32, 33, 37, 38, 49, 63, 74

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–222, 2011. 84

L. M. Briceño-Arias and P. L. Combettes. Convex variational formulation with smooth coupling for multicomponent signal decomposition and recovery. *Numerical Mathematics Theory Methods and Applications*, 2:485–508, 2009. 22

L. M. Briceño-Arias, P. L. Combettes, J.-C. Pesquet, and N. Pustelnik. Proximal algorithms for multicomponent image recovery problems. *Journal of Mathematical Imaging and Vision*, 41(1–2):3–22, 2011. 72

L. M. Briceño-Arias, G. Chierchia, E. Chouzenoux, and J.-C. Pesquet. A random block-coordinate Douglas–Rachford splitting method with low computational complexity for binary logistic regression. *Computational Optimization and Applications*, pages 1–20, 2017. 41

A. S. Brito, J. X. da Cruz Neto, J. O. Lopes, and P. R. Oliveira. Interior proximal algorithm for quasiconvex programming problems and variational inequalities with linear constraints. *Journal of Optimization Theory and Applications*, 154(1):217–234, 2012. 47

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov chain Monte Carlo*. CRC press, Boca Raton, USA, 2011. 13

A. Cabot. Proximal point algorithm controlled by a slowly vanishing term: applications to hierarchical minimization. *SIAM Journal on Optimization*, 15(2):555–572, 2005. 47

J.-F. Cai, R. H. Chan, and M. Nikolova. Fast two-phase image deblurring under impulse noise. *Journal of Mathematical Imaging and Vision*, 36(1): 46, 2010. 10

X. Cai, R. Chan, M. Nikolova, and T. Zeng. A three-stage approach for segmenting degraded color images: Smoothing, lifting and thresholding (SLaT). *Journal of Scientific Computing*, 72(3):1313–1332, 2017. 123

L. Calatroni and A. Chambolle. Backtracking strategies for accelerated descent methods with smooth composite objectives. *SIAM Journal on Optimization*, 29(3):1772–1798, 2019. 135

C. W. Carroll. The created response surface technique for optimizing nonlinear, restrained systems. *Operations Research*, 9(2):169–184, 1961. 33

G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992. 14

A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004. 11

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. 28, 29, 139

R. H. Chan, K. K. Kan, M. Nikolova, and R. J. Plemmons. A two-stage method for spectral-spatial classification of hyperspectral images. *arXiv preprint arXiv:1806.00836*, 2018. 67

T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 41

C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs. A variational formulation for frame-based inverse problems. *Inverse Problems*, 23(4): 1495, 2007. 22, 23, 24

C. Chaux, A. Benazza-Benyahia, J.-C. Pesquet, and L. Duval. Wavelet transform for the denoising of multivariate images. In C. Collet, J. Chanussot, and K. Chehdi, editors, *Multivariate Image Processing*, pages 203–237. ISTE Ltd and John Wiley & Sons Inc, Washington, Hoboken, USA, 2010. 48

C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016. 31

G. H. G. Chen and R. T. Rockafellar. Convergence rates in forward–backward splitting. *SIAM Journal on Optimization*, 7(2):421–444, 1997. 26

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 16

Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017. 17

J.-T. Chien and C.-H. Lee. Deep unfolding for topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):318–331, 2018. 17

E. Chouzenoux, S. Moussaoui, and J. Idier. Majorize–minimize linesearch for inversion methods involving barrier function optimization. *Inverse Problems*, 28(6):065011, 2012. 42

E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot. A majorize-minimize subspace approach for ℓ_2-ℓ_0 image regularization. *SIAM Journal on Imaging Sciences*, 6(1):563–591, 2013a. 32

E. Chouzenoux, F. Zolyniak, E. Gouillart, and H. Talbot. A majorize-minimize memory gradient algorithm applied to X-ray tomography. In *Proc. of the IEEE International Conference on Image Processing*, pages 1011–1015, Melbourne, Australia, 15–18 Sep 2013b. 73

E. Chouzenoux, M. Legendre, S. Moussaoui, and J. Idier. Fast constrained least squares spectral unmixing using primal-dual interior-point optimization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(1):59–69, 2014a. 41, 67, 69

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, 2014b. 19, 21, 26, 42, 43, 113, 135

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function. *Journal of Optimization Theory and Applications*, 162(1):107–132, 2014c. 75

E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot. A convex approach for image restoration with exact Poisson–Gaussian likelihood. *SIAM Journal on Imaging Sciences*, 8(4):2662–2682, 2015. 10, 136

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. A block coordinate variable metric forward–backward algorithm. *Journal of Global Optimization*, 66 (3):457–485, 2016. 31, 46

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, New York, USA, 2011. 41, 113, 116

P. L. Combettes and J.-C. Pesquet. Deep neural network structures solving variational inequalities. *arXiv preprint arXiv:1808.07526*, 2018. 95, 96, 98

P. L. Combettes and B. C. Vũ. Variable metric forward–backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9): 1289–1318, 2014. 21, 26, 27, 43, 139

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005. 115

P. L. Combettes, D. Dũng, and B. C. Vũ. Proximity for sums of composite functions. *Journal of Mathematical Analysis and Applications*, 380(2): 680–688, 2011. 28, 139

P. L. Combettes, L. Condat, J.-C. Pesquet, and B. C. Vũ. A forward–backward view of some primal-dual optimization methods in image recovery. In *Proc. of the IEEE International Conference on Image Processing*, pages 4141–4145, Paris, France, 27–30 Oct 2014. 69

M. L. Comer and E. J. Delp. The EM/MPM algorithm for segmentation of textured images: analysis and further experimental results. *IEEE Transactions on Image Processing*, 9(10):1731–1744, 2000. 111

L. Condat. A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013. 28, 29, 139

J. Corander, M. Gyllenberg, and T. Koski. Bayesian model learning based on a parallel MCMC strategy. *Statistics and Computing*, 16(4):355–362, 2006. 12

M.-O. Czarnecki, N. Noun, and J. Peypouquet. Splitting forward-backward penalty scheme for constrained variational problems. *Journal of Convex Analysis*, 23(2):531–565, 2016. 85

W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009. 10

E. De Vito, L. Rosasco, A. Caponnetto, U. De Giovannini, and F. Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6(May):883–904, 2005. 9

C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré. Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection. *SIAM Journal on Imaging Sciences*, 7(4):2448–2487, 2014. 11

S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017. 17, 82

N. Dobigeon, A. Basarab, D. Kouamé, and J.-Y. Tourneret. Regularized Bayesian compressed sensing in ultrasound imaging. In *Proc. of the European Signal Processing Conference*, pages 2600–2604, Bucharest, Romania, 27–31 Aug 2012. IEEE. 111

N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero. Nonlinear unmixing of hyperspectral images: Models and algorithms. *IEEE Signal Processing Magazine*, 31(1):82–94, 2013. 10

A. L. Dontchev and R. T. Rockafellar. Implicit functions and solution mappings. *Springer Monogr. Math.*, 2009. 88, 91

C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating second-order functional knowledge for better option pricing. In *Proc. of Advances in Neural Information Processing Systems*, pages 472–478, Vancouver, Canada, 3–8 Dec 2001. 93

S. Durand and M. Nikolova. Stability of minimizers of regularized least squares objective functions I: study of the local behaviour. *Applied Mathematics and Optimization (Springer-Verlag New York)*, 53:185–208, 2006. 84

S. Durand, J. Fadili, and M. Nikolova. Multiplicative noise removal using ℓ_1 fidelity on frame coefficients. *Journal of Mathematical Imaging and Vision*, 36(3):201–226, 2010. 10

A. Durmus, E. Moulines, and M. Pereyra. Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018. 113

J. Eckstein and D. P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992. 30

H. Erdogan and J. A. Fessler. Monotonic algorithms for transmission tomography. *IEEE Transactions on Medical Imaging*, 18(9):801–814, 1999. 32

A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM review*, 44(4):525–597, 2002. 32, 33

M. Fortin and R. Glowinski. *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, volume 15. Elsevier, Amsterdam, Netherlands, 2000. 30

K. Fountoulakis and J. Gondzio. Performance of first- and second-order methods for ℓ_1-regularized least squares problems. *Computational Optimization and Applications*, 65(3):605–635, 2016. 41

P. Frankel, G. Garrigos, and J. Peypouquet. Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *Journal of Optimization Theory and Applications*, 165(3):874–900, 2015. 26, 42, 59

J. Frecon, N. Pustelnik, H. Wendt, L. Condat, and P. Abry. Multifractal-based texture segmentation using variational procedure. In *Proc. of the IEEE Image, Video, and Multidimensional Signal Processing Workshop*, pages 1–5, Bordeaux, France, 11–12 Jul 2016. 71

H. Fu, M.K. Ng, M. Nikolova, and J.L. Barlow. Efficient minimization methods of mixed $\ell 2$-$\ell 1$ and $\ell 1$-$\ell 1$ norms for image restoration. *SIAM Journal on Scientific Computing*, 27(6):1881–1902, 2006. 41, 48

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. 30

G. Garrigos, L. Rosasco, and S. Villa. Iterative regularization via dual diagonal descent. *Journal of Mathematical Imaging and Vision*, 60(2):189–215, 2018. 46

D. Ge, H. Wang, Z. Xiong, and Y. Ye. Interior-point methods strike back: Solving the Wasserstein barycenter problem. *arXiv preprint arXiv:1905.12895*, 2019. 32

A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. *Bayesian Statistics*, 5(599-608):42, 1996. 13

P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. On projected Newton barrier methods for linear programming and an equivalence to KarmarkarâĂŹs projective method. *Mathematical Programming*, 36(2):183–209, 1986. 32

J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012. 32, 38

J. Gondzio and A. Grothey. Direct solution of linear systems of size 10^9 arising in optimization with interior point methods. In *Proc. of the International Conference on Parallel Processing and Applied Mathematics*, pages 513–525, Poznan, Poland, 11–14 Sep 2005. Springer. 32

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, USA, 2016. http://www.deeplearningbook.org. 15, 16, 103

E. Gouillart, F. Krzakala, M. Mézard, and L. Zdeborová. Belief propagation reconstruction for discrete tomography. *Inverse Problems*, 29(3):035003, 2013. 73

K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proc. of the International Conference on Machine Learning*, pages 399–406, Haifa, Israel, 21–24 Jun 2010. Omnipress. 17

J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton university bulletin*, pages 49–52, 1902. 10

R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):768–804, 1979. 72

S. Harizanov, J.-C. Pesquet, and G. Steidl. Epigraphical projection for solving least squares Anscombe transformed constrained optimization problems. In *Proc. of the International Conference on Scale Space and Variational Methods in Computer Vision*, pages 125–136, Leibnitz, Austria, 2–6 Jun 2013. Springer. 44

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer-Verlag, New York, USA, 2009. 48

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 13

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, USA, 26 Jun– 1 Jul 2016. 16

J. R. Hershey, J. Le Roux, and F. Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014. 17, 81, 92

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms*. Springer-Verlag, Berlin, Heidelberg, Germany, 1996. 22

M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, Berlin, Heidelberg, Germany, 1990. 16

Y. Huang and Y. Dong. New properties of forward–backward splitting and a practical proximal-descent algorithm. *Applied Mathematics and Computation*, 237:60–68, 2014. 52, 55

D. R. Hunter and K. Lange. Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77, 2000. 32

D. R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004. 31

O. Husby and H. Rue. Estimating blood vessel areas in ultrasound images using a deformable template model. *Statistical Modelling*, 4(3):211–226, 2004. 111

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 15

M.-D. Iordache, J. M. Bioucas-Dias, and A.o Plaza. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4484–4502, 2012. 10, 31, 67, 69, 76

A. N. Iusem. On the convergence properties of the projected gradient method for convex optimization. *Computational Applied Mathematics*, 22(1):37–52, 2003. 84, 103

A.N. Iusem, B.F. Svaiter, and M. Teboulle. Entropy-like proximal methods in convex programming. *Mathematics of Operations Research*, 19(4):790–814, 1994. 47

M. W. Jacobson and J. A. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Transactions on Image Processing*, 16(10):2411–2422, 2007. 32

F. Jarre. Interior-point methods for convex programming. *Applied Mathematics and Optimization*, 26(3):287–311, 1992. 38

R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proc. of the International Conference on Machine Learning*, volume 1, page 2, Haifa, Israel, 21–24 June 2010. Citeseer. 9

J. A. Jensen. A model for the propagation and scattering of ultrasound in tissue. *The Journal of the Acoustical Society of America*, 89(1):182–190, 1991. 111

J. A. Jensen. Deconvolution of ultrasound images. *Ultrasonic imaging*, 14 (1):1–15, 1992. 112

J. A. Jensen. Simulation of advanced ultrasound systems using Field II. In *Proc. of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 636–639, Arlington, USA, 15–18 April 2004. 121

J. A. Jensen, J. Mathorne, T. Gravesen, and B. Stage. Deconvolution of in-vivo ultrasound B-mode images. *Ultrasonic Imaging*, 15(2):122–133, 1993. 112

K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017. 16

K. Jittorntrum. Solution point differentiability without strict complementarity in nonlinear programming. In *Sensitivity, Stability and Parametric Analysis*, pages 127–138. Springer, Berlin, Heidelberg, Germany, 1984. 36

C. A. Johnson, J. Seidel, and A. Sofer. Interior-point methodology for 3-D PET reconstruction. *IEEE Transactions on Medical Imaging*, 19(4): 271–285, 2000. 41

A. C. Kak and M. Stanley. *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia, USA, 2001. 72

U. S. Kamilov and H. Mansour. Learning optimal nonlinearities for iterative thresholding algorithms. *IEEE Signal Processing Letters*, 23(5):747–751, 2016. 17

A. Kaplan and R. Tichatschke. Proximal methods in view of interior-point strategies. *Journal of Optimization Theory and Applications*, 98(2):399–429, 1998. 42, 84, 85, 139

N. Karmarkar. A new polynomial-time algorithm for linear programming. In *Proc. of the Annual ACM symposium on Theory of Computing*, pages 302–311, Washington, USA, 30 Apr–2 May 1984. ACM. 32

N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, 2002. 67

R. A. Ketcham and W. D. Carlson. Acquisition, optimization and interpretation of X-ray computed tomographic imagery: applications to the geosciences. *Computers & Geosciences*, 27(4):381–400, 2001. 73

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, 2007. 32, 41

S. Kindermann, S. Osher, and P. W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation*, 4(4): 1091–1115, 2005. 137

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 102

M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *Proc. of the International Conference on Computer Vision*, pages 2236–2243. IEEE, 6–13 Nov 2011. 11, 44

N. Komodakis and J.-C. Pesquet. Playing with duality: An overview of recent primal–dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, 32(6):31–54, 2015. 30, 69, 84

J. Kowalik. Nonlinear programming procedures and design optimization. *ACTA Polytechnica Scandinavica, Mathematics and Computing Machinery Series*, (13):47, 1966. 33

P. T. Krasopoulos and N. G. Maratos. *An Interior Point Recurrent Neural Network for Convex Optimization Problems*, pages 409–427. Springer New York, New York, USA, 2014. 82

S. Krishnan, K. W. Rigby, and M. O'donnell. Improved estimation of phase aberration profiles. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 44(3):701–713, 1997. 123

K. Kurdyka. On gradients of functions definable in o-minimal structures. In *Annales de l'Institut Fourier*, volume 48, pages 769–783, 1998. 19

R. L. Lagendijk and J. Biemond. *The Handbook of Image and Video Processing*. Academic Press, Cambridge, USA, 2005. 83

C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 105–114, Honolulu, USA, 21–26 Jul 2017. 16

N. Lermé, F. Malgouyres, D. Hamoir, and E. Thouin. Bayesian image restoration for mosaic active imaging. *Inverse Problems and Imaging* , 8(3):733–760, 2014. URL https://hal.archives-ouvertes.fr/hal-00758753. 12

T. Lesieur, F. Krzakala, and L. Zdeborová. MMSE of probabilistic low-rank matrix estimation: Universality with respect to the output channel. In *Proc. of the Annual Allerton Conference on Communication, Control, and Computing*, pages 680–687, Allerton, USA, 30 Sep–2 Oct 2015. IEEE. 12

A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, Miami, USA, 20–25 Jun 2009. IEEE. 10, 101

G. Li and T.K. Pong. Calculus of the exponent of Kurdyka–Łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of Computational Mathematics*, 18(5):1199–1232, 2018. 20, 21, 59, 141

K. Li and J. Malik. Learning to optimize. *arXiv preprint arXiv:1606.01885*, 2016. 82

Z. Li, F. Malgouyres, and T. Zeng. Regularized non-local total variation and application in image restoration. *Journal of Mathematical Imaging and Vision*, 59(2):296–317, 2017. 137

S. Lojasiewicz. *Une propriété topologique des sous-ensembles analytiques réels.* Éditions du Centre National de la Recherche Scientifique, Paris, France, 1963. 19

C. López-Martínez and X. Fabregas. Polarimetric SAR speckle noise model. *IEEE Transactions on Geoscience and Remote Sensing*, 41(10):2232–2242, 2003. 10

L. B. Lucy. Astronomical inverse problems. *Reviews in Modern Astronomy*, 7:31–50, 1994. 9

F. Malgouyres. Minimizing the total variation under a general convex constraint for image restoration. *IEEE Transactions on Image Processing*, 11 (12):1450–1456, 2002. 41

F. Malgouyres and J. Landsberg. Multilinear compressive sensing and an application to convolutional linear networks. *SIAM Journal on Mathematics of Data Science*, 1(3):446–475, 2019. 94

S. Mallat. *A wavelet tour of signal processing.* Elsevier, 1999. 93, 100, 104, 123

M. Mardani, H. Monajemi, V. Papyan, S. Vasanawala, D. Donoho, and J. Pauly. Recurrent generative adversarial networks for proximal learning and automated compressive image recovery. *arXiv preprint arXiv:1711.10046*, 2017. 17, 82

Y. Marnissi, E. Chouzenoux, A. Benazza-Benyahia, and J.-C. Pesquet. Majorize-minimize adapted Metropolis-Hastings algorithm. *HAL preprint HAL:01909153*, 2018. 114

M. T. McCann, K. H. Jin, and M. Unser. Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Processing Magazine*, 34(6):85–95, 2017. 15, 16

T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proc. of the IEEE International Conference on Computer Vision*, pages 1781–1790, Venice, Italy, 22–29 Oct 2017. 104

W. Menke. *Geophysical data analysis: Discrete inverse theory.* Academic press, Cambridge, USA, 2018. 9

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. 13

O. Michailovich and A. Tannenbaum. Blind deconvolution of medical ultrasound images: A parametric inverse filtering approach. *IEEE Transactions on Image Processing*, 16(12):3005–3019, 2007. 112

J.-J. Moreau. Proximité et dualité dans un espace Hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965. 21

J.-J. Moreau. Fonctionnelles convexes. *Séminaire Jean Leray*, (2):1–108, 1966. 19

S. M. Moser. Capacity results of an optical intensity channel with input-dependent Gaussian noise. *IEEE Transactions on Information Theory*, 58(1):207–223, 2012. 10

S. Moussaoui, D. Brie, A. Mohammad-Djafari, and C. Carteret. Separation of non-negative mixture of non-negative sources using a bayesian approach and MCMC sampling. *IEEE Transactions on Signal Processing*, 54(11): 4133–4145, 2006. 12

O. Musse, F. Heitz, and J.-P. Armspach. Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Transactions on Image Processing*, 10(7):1081–1093, 2001. 11, 44

A. Nair, B. D. Kuban, E. M. Tuzcu, P. Schoenhagen, S. E. Nissen, and D. G. Vince. Coronary plaque classification with intravascular ultrasound radiofrequency data analysis. *Circulation*, 106(17):2200–2206, 2002. 111

R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011. 113

Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, Philadelphia, USA, 1994. 32

Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Doklady Akademii Nauk*, 269(3):543–547, 1983. 135

J. Ng, R. Prager, N. Kingsbury, G. Treece, and A. Gee. Modeling ultrasound imaging as a linear, shift-variant system. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 53(3):549–563, 2006. 111, 112

M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120, 2004. 41

A. Noble and D. Boukerroui. Ultrasound image segmentation: a survey. *IEEE Transactions on Medical Imaging*, 25(8):987–1010, 2006. 111

J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Adaptive total variation image deblurring: a majorization–minimization approach. *Signal processing*, 89(9):1683–1693, 2009. 10

J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Academic Press, New York, USA, 1970. 31

S. Osher, A. SolÃę, and L. Vese. Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Modeling & Simulation*, 1(3):349–370, 2003. 72

N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 121, 123

S. K. Pal and S. Mitra. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, 3(5):683–697, 1992. 16

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. De Vito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *Proc. of the NIPS Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques*, Long Beach, CA, USA, 9 Dec 2017. 94, 103

M. Pereyra. Proximal Markov chain Monte Carlo algorithms. *Statistics and Computing*, 26(4):745–760, 2016. 15, 113, 114, 115, 137

M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. O. Hero, and S. McLaughlin. A survey of stochastic simulation and optimization methods in signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):224–241, 2016. 13

M. A. Pereyra, N. Dobigeon, H. Batatia, and J.-Y. Tourneret. Segmentation of skin lesions in 2D and 3D ultrasound images using a spatially coherent generalized Rayleigh mixture model. *IEEE Transactions on Medical Imaging*, 31(8):1509–1520, 2012. 111, 129

G. Peyré, S. Bougleux, and L. Cohen. Non-local regularization of inverse problems. In *Proc. of the European Conference on Computer Vision*, pages 57–68, Marseille, France, 12–18 Oct 2008. Springer. 137

A. Pirayre, Y. Zheng, L. Duval, and J.-C. Pesquet. HOGMep: Variational Bayes and higher-order graphical models applied to joint image recovery and segmentation. In *Proc. of the IEEE International Conference on Image Processing*, pages 3775–3779, Beijing, China, 17–20 Sep 2017. 112

M. Ploquin, A. Basarab, and D. Kouamé. Resolution enhancement in medical ultrasound imaging. *Journal of Medical Imaging*, 2(1):017001, 2015. 111

S. Poddar and M. Jacob. Dynamic MRI using smoothness regularization on manifolds (SToRM). *IEEE Transactions on Medical Imaging*, 35(4): 1106–1115, 2015. 11

N. Pustelnik and L. Condat. Proximity operator of a sum of functions; application to depth map estimation. *IEEE Signal Processing Letters*, 24 (12):1827–1831, 2017. 115

N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet. Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, pages 1–34, 1999. 48, 67

N. Pustelnik, H. Wendt, and P. Abry. Local regularity for texture segmentation: Combining wavelet leaders and proximal minimization. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5348–5352, Vancouver, Canada, 26–31 May 2013. 72

N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet. *Wavelet-based image deconvolution and reconstruction*. John Wiley & Son, Hoboken, USA, February 2016. doi: 10.1002/047134608X.W8294. 11

E. A. P. Quiroz, L. M. Ramirez, and P. R. Oliveira. An inexact proximal method for quasiconvex minimization. *European Journal of Operational Research*, 246(3):721–729, 2015. 47

H. Raguet and L. Landrieu. Preconditioning of a generalized forward-backward splitting and application to optimization on graphs. *SIAM Journal on Imaging Sciences*, 8(4):2706–2739, 2015. 69

H. Raguet, J. Fadili, and G. Peyré. A generalized forward–backward splitting. *SIAM Journal on Imaging Sciences, Society for Industrial and Applied Mathematics*, 6(3):1199–1226, 2013. 27, 69, 139

A. Ramadhan, F. Mahmood, and A. Elci. Image denoising by median filter in wavelet domain. *arXiv preprint arXiv:1703.06499*, 2017. 93

C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, Berlin, Germany, 2013. 13, 14

C. P. Robert, V. Elvira, N. Tawn, and C. Wu. Accelerating MCMC algorithms. *Wiley Interdisciplinary Reviews: Computational Statistics*, 10(5), 2018. 15, 113

G. O. Roberts and O. Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4(4):337–357, 2002. 113

G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. 114

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997. 13, 14, 139

R. T. Rockafellar. *Convex analysis*, volume 28. Princeton University Press, Princeton, USA, 1970. 19

L. Rosasco and S. Villa. Learning with incremental iterative regularization. In *Proc. of Advances in Neural Information Processing Systems*, pages 1630–1638, Montréal, Canada, 7–12 Dec 2015. 12, 17

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992. 11, 48

J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett. Poisson noise reduction with non-local PCA. *Journal of Mathematical Imaging and Vision*, 48(2):279–294, 2014. 10

S. Salzo. The variable metric forward-backward splitting algorithm under mild differentiability assumptions. *SIAM Journal on Optimization*, 27(4): 2153–2181, 2017. 43, 46, 52, 53

S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex analysis*, 19(4):1167–1192, 2012. 135

O. Scherzer. The use of Morozov's discrepancy principle for Tikhonov regularization for solving nonlinear ill-posed problems. *Computing*, 51(1): 45–60, 1993. 11

E. D. Schifano, R. L. Strawderman, and M. T. Wells. Majorization-minimization algorithms for nonsmoothly penalized objective functions. *Electronic Journal of Statistics*, 4:1258–1299, 2010. 119

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 15

U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781, Columbus, USA, 24–27 Jun 2014. 17

A. Schreck, G. Fort, S. Le Corff, and E. Moulines. A shrinkage-thresholding Metropolis adjusted Langevin algorithm for Bayesian variable selection. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):366–375, 2016. 113

C. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf. A machine learning approach for non-blind image deconvolution. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1074, Portland, USA, 23–28 Jun 2013. IEEE. 16, 103

C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7): 1439, 2016. 16

I. Selesnick. Total variation denoising (an MM algorithm). *NYU Polytechnic School of Engineering Lecture Notes*, 2012. 32

S. Setzer, G. Steidl, and T. Teuber. Deblurring Poissonian images by split Bregman techniques. *Journal of Visual Communication and Image Representation*, 21(3):193–199, 2010. 69

R. Shefi and M. Teboulle. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, 24(1):269–297, 2014. 69

G. Sonnevend. An "analytical centre" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In *System Modelling and Optimization*, pages 866–875. Springer, Berlin, Heidelberg, Germany, 1986. 38

S. Sotthivirat and J. A. Fessler. Image recovery using partitioned-separable paraboloidal surrogate coordinate ascent algorithms. *IEEE Transactions on Image Processing*, 11(3):306–317, 2002. 32

A. M. Stuart, J. Voss, and P. Wilberg. Conditional path sampling of SDEs and the Langevin MCMC method. *Communications in Mathematical Sciences*, 2(4):685–697, 2004. 113, 114

S. Sudha, G. R. Suresh, and R. Sukanesh. Speckle noise reduction in ultrasound images by wavelet thresholding based on weighted variance. *International Journal of Computer Theory and Engineering*, 1(1):7, 2009. 10

J. Sun and Z. Xu. Color image denoising via discriminatively learned iterative shrinkage. *IEEE Transactions on Image Processing*, 24(11):4148–4159, 2015. 17

J. Sun, H. Li, Z. Xu, and Y. Yang. Deep ADMM-Net for compressive sensing MRI. In *Proc. of Advances in Neural Information Processing Systems*, pages 10–18, Barcelona, Spain, 5–10 Dec 2016. 17

T. L. Szabo. *Diagnostic ultrasound imaging: inside out.* Academic Press, Cambridge, USA, 2004. 111

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 16, 95

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. 11

T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *Proc. of the IEEE International Conference on Computer Vision*, pages 4799–4807, Venice, italy, 22–29 Oct 2017. 16

T. B. Trafalis, T. A. Tutunji, and N. P. Couëllan. Interior point methods for supervised training of artificial neural networks with bounded weights. In P. M. Pardalos, D. W. Hearn, and W. W. Hager, editors, *Network Optimization*, pages 441–470, Berlin, Heidelberg, Germany, 1997. Springer. 82

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009. 42

Paul Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38(2): 431–446, 2000. 26

T. Valkonen. Interior-proximal primal-dual methods. *arXiv preprint arXiv:1706.07067*, 2017. 42

L. van den Dries, A. Macintyre, and D. Marker. The elementary theory of restricted analytic fields with exponentiation. *Annals of Mathematics*, 140 (1):183–205, 1994. 19

C. R. Vogel. *Computational methods for inverse problems*, volume 23. SIAM, Philadelphia, USA, 2002. 93

B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3): 667–681, 2013. 28, 29, 139

S. Wang, S. Fidler, and R. Urtasun. Proximal deep structured models. In *Proc. of Advances in Neural Information Processing Systems*, pages 865–873, Barcelona, Spain, 5–10 Dec 2016. 17

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 102, 103, 123

M. H. Wright. Interior methods for constrained optimization. *Acta numerica*, 1:341–407, 1992. 32, 34, 36

L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proc. of the European Conference on Computer Vision*, pages 157–170. Springer, 5–11 Sep 2010. 83

L. Xu, J. S. J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Proc. of Advances in Neural Information Processing Systems*, pages 1790–1798, Montréal, Canada, 8–13 Dec 2014a. 16, 101

Y. Xu, Y. Kabashima, and L. Zdeborová. Bayesian signal reconstruction for 1-bit compressed sensing. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(11):P11015, 2014b. 12

J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11): 2861–2873, 2010. 9

Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007. 12

J. Zhang and B. Ghanem. ISTA-Net: Interpretable optimization–inspired deep network for image compressive sensing. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1828–1837, Salt Lake City, USA, 18–23 Jun 2018. 17

J. Zhang, J. Pan, W.-S. Lai, R. W. H. Lau, and M.-H. Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3817–3825, Honolulu, HW, USA, 21–26 Jul 2017a. URL https://github.com/zhjwustc/cvpr17_iter_deblur_testing_matconvnet. 104

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017b. 16

K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. In *Proc. of the IEEE Conference on Computer*

Vision and Pattern Recognition, volume 2, Honolulu, HW, USA, 21–26 Jul 2017c. 100, 103

Q. Zhang, Y. Nian Wu, and S.-C. Zhu. Interpretable convolutional neural networks. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, Salt Lake City, USA, 18–23 Jun 2018. 94

N. Zhao, A. Basarab, D. Kouamé, and J.-Y. Tourneret. Joint segmentation and deconvolution of ultrasound images using a hierarchical Bayesian model based on generalized Gaussian priors. *IEEE Transactions on Image Processing*, 25(8):3736–3750, 2016. 112, 116, 122

L. Zhu, E. Liu, and J. H. McClellan. Seismic data denoising through multiscale and sparsity-promoting dictionary learning. *Geophysics*, 80(6):45–57, 2015. 11

D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proc. of the IEEE International Conference on Computer Vision*, pages 479–486. IEEE, 6–13 Nov 2011. 103

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: series B (Statistical Methodology*, 67(2):301–320, 2005. 11