Multilevel proximal methods and application to image restoration

Guillaume Lauga

THÈSE

pour l'obtention du grade de Docteur, délivré par

l'ÉCOLE NORMALE SUPÉRIEURE DE LYON

Discipline : **MATHÉMATIQUES**

École Doctorale N°512
InfoMaths - Informatique et de Mathématiques

Présentée et soutenue publiquement le 17 décembre 2024, par :
Guillaume LAUGA

Méthodes proximales multi-niveaux et application à la restauration d'images

Multilevel proximal methods and application to image restoration

---

Directeur de Thèse : Paulo Gonçalves

Devant le jury composé de :

Aude Rondepierre, Professeure des universités, *INSA Toulouse*	Rapporteure
Ivan Selesnick, Professeur, *New York University, Tandon School of Engineering*	Rapporteur
Panos Parpas, Personnalité scientifique, *Imperial College London*	Examinateur
Jean-Christophe Pesquet, Professeur des universités, *CentraleSupélec*	Examinateur
Bruno Torresani, Professeur des universités, *Université d'Aix-Marseille*	Examinateur
Elisa Riccietti, Maitresse de conférence, *ENS Lyon*	Examinatrice
Nelly Pustelnik, Directrice de Recherche, *CNRS*	Examinatrice
Paulo Gonçalves, Directeur de Recherche, *Inria*	Directeur de thèse

Contents

Remerciements

J'écris ces remerciements bien après la soutenance. Parce qu'elle s'est si bien passée. Parce que je veux pouvoir remercier à leur juste valeur celles et ceux qui m'ont permis d'en arriver là. Les émotions qui m'ont traversées à la fin, ont été l'expression directe de ma gratitude envers les personnes qui m'ont entouré, accompagné, et soutenu tout au long de ces trois années de thèse à l'ENS.

Pour commencer, je veux remercier mes trois encadrant et encadrantes de thèse, Paulo, Nelly et Elisa. Mes recherches pour trouver une thèse ont été chaotiques, et pour je ne sais quelle raison, j'ai su immédiatement lorsque nous avons discuté pour la première fois que vous étiez les trois personnes avec qui je devais faire ma thèse. À tel point que j'ai appelé un samedi Elisa - qui a mis son numéro sur son site - après les circonvolutions dont j'ai l'habitude pour pouvoir confirmer mon choix. Et c'est allé au-delà de toutes mes espérances, du premier au dernier jour de ces trois ans.

Tout d'abord par votre soutien scientifique. Vos trois points de vue, vos trois visions m'ont permis en tout instant de compléter autant que possible mes recherches et mon travail, qui sont partis de connaissances frémissantes à une maîtrise de ce qu'il m'a été nécessaire pour achever cette thèse.

Avant tout parce que grâce à vous, ces trois années ont été formidables, et le mot est faible ici. Je crois que je n'aurais pas pu accomplir tout le travail qu'a représenté cette thèse si je n'avais pas su que je pouvais compter sur vous pour que je puisse tirer le maximum de chaque jour. J'ai fait beaucoup de pas de côté, volontaires pour certains - notamment mes choix « artistiques » -, involontaires pour d'autres, mais vous avez toujours su me guider dans une direction qui est la bonne.

Ce n'est pas tous les jours que l'on peut être soi sans en douter, et je vous en dois beaucoup. Mille mercis.

Je veux aussi remercier les membres du jury. Merci beaucoup Aude Rondepierre d'avoir accepté de rapporter ma thèse, et d'y avoir accordé plus d'attention que je ne pouvais espérer. Thanks a lot Ivan Selesnick for also accepting to report my thesis. I was also delighted that Panos Parpas, whose work helped me tremendously at the beginning of my thesis, accepted to be part of this jury! It was a pleasure meeting you after these three years. Merci beaucoup Bruno (en compagnie de Bora) de m'avoir suivi épisodiquement durant cette thèse, et d'avoir été là pour cette conclusion ! Enfin je veux remercier Jean-Christophe Pesquet d'avoir accepté de présider ce jury, c'était un honneur pour moi.

J'ai eu l'occasion pendant ma thèse d'entamer plusieurs collaborations qui ont non seulement enrichi ma thèse scientifiquement mais aussi humainement. Merci Audrey - de m'avoir fait confiance - et Yves de m'avoir permis de travailler sur la radio-interférométrie, et aussi de m'avoir fait découvrir Edinburgh ! Merci aussi Luis, pour tes remarques et suggestions précieuses, qui nous ont mené à ce dernier chapitre, chapitre dont je suis le plus fier sur le plan théorique.

Merci aussi à tous les membres permanents de l'équipe, Rémi G, tes conseils et ton entrain m'ont toujours été très précieux; Mathurin, pour ton soutien indéfectible contre vents, énergies, et marées; Titouan, pour ton humour et tes bonnes paroles quotidiennes ; Pascal, pour ta bonne humeur, tes blagues que je suis rapide à comprendre mais qu'il faut m'expliquer longtemps, et tes vraies fausses informations ; Marion et Simon, pour les bons moments et discussions qu'on a partagés. Merci à vous.

Cette thèse je l'ai commencée avec six autres doctorants désormais docteurs, Clément, Samir, Anthony, Tung, Léon, Antoine, que j'ai apprécié dès le premier jour. Vous m'avez inspiré à donner le meilleur de moi-même. Vous avez aussi fait de cette thèse, démarré comme une aventure solitaire, un travail d'équipe. Pour ça, je ne pourrai vous remercier assez !

Merci aussi à toutes celles et ceux que j'ai croisé au cours de ces trois années à l'ENS et ailleurs ! À tous les anciens et affiliés de Dante et Ockham, Sybille, Wassim, Rémi V, Ayoub, Badr, Meriem, Esther, Luc, Solène, Valérie, merci beaucoup ! J'ai eu grand plaisir à vous côtoyer, et j'en aurai encore plus si on se recroise ! Merci Clara, merci Gabriel, votre bonne humeur et vos anecdotes outre-atlantiques vont me manquer. Merci Marie pour ces conférences qu'on a partagées, et à ta family ! Merci beaucoup Myriam, nos sessions hebdomadaires d'escalades m'ont et vont me manquer.

Aux collègues du laboratoire de Physique, Léo, Victor, Nils, Juliana, Guillermo, Julian, merci à vous, ce fut un plaisir ! Merci aussi aux MALIP, au CBP, à Patrice, à Diane, et aux autres résidents du M7-1H, ce couloir va bien me manquer !

Merci à vous tous pour tout ce qu'on a partagé à Lyon en et hors du bureau

Merci beaucoup aux nouveaux Arthur, Maël et mon « remplaçant » Edgar. L'ambiance de l'équipe ces derniers mois n'auraient pas été la même sans vous ! Je vous souhaite de vivre votre thèse à fond, et de la terminer dans la même joie que celle que j'ai vécue.

Enfin je veux remercier mes camarades de bureau de ces quasi deux dernières années, Anne, Can et Étienne.

Merci beaucoup Étienne, tu as mis le rythme de chaque journée par nos discussions, nos cafés. Charles Pasco t'en doit une.

Merci beaucoup Can, tu es une des plus belles personnes que je connaisse, et ce fut un vrai plaisir de faire un bout de chemin avec toi !

Merci beaucoup Anne, sans toi cette dernière année n'aurait pas été pareille dans son déroulement et son dénouement.

Avec vous tous, mon Rambouillet est au complet.

Pour terminer, le plus grand des mercis je le réserve à mes amis, et famille, qui depuis tant d'années sont à mes côtés.

Abstract

The size of image restoration problems is constantly increasing. This growth poses a major scaling problem for optimization algorithms, which struggle to provide satisfactory solutions in a reasonable amount of time.

Among the methods proposed to overcome this challenge, multilevel methods seem to be an ideal candidate. By systematically reducing the size of the problem, the computational cost of solving it can be drastically decreased. This type of approach is standard in the numerical solution of partial differential equations (PDEs), with theoretical guarantees and practical demonstrations to explain their success.

However, current multilevel optimization methods do not have the same guarantees nor the same performance. In this thesis, we propose to bridge a part of this gap by introducing a new multilevel algorithm, IML FISTA, which has the optimal theoretical convergence guarantees for convex non-smooth optimization problems, i.e. convergence to a minimiser and convergence rate of the objective function to a minimum value. IML FISTA is also able to handle state-of-the-art regularizations in image restoration.

By comparing IML FISTA with standard algorithms on many image restoration problems: deblurring, denoising, reconstruction of missing pixels for colour and hyperspectral images, and reconstruction of radio-interferometric images, we show that IML FISTA is capable of significantly speeding up the resolution of these problems. As IML FISTA's framework is sufficiently general, it can be adapted to many other image restoration problems.

We conclude this thesis by proposing a new point of view on multilevel algorithms, by demonstrating their equivalence, in certain cases, with coordinate descent algorithms, which are much more widely studied in the non-smooth optimization literature. This new theoretical framework allows us to analyse multilevel algorithms more rigorously, and in particular to extend their convergence guarantees to non-smooth and non-convex problems. This framework is less general than that of IML FISTA, but it paves the way for a more theoretically robust design of multilevel algorithms.

Résumé

La taille des problèmes de restauration d'images ne fait qu'augmenter. Cette croissance pose un problème majeur de passage à l'échelle pour les algorithmes d'optimisation, qui peinent à fournir des solutions satisfaisantes en un temps raisonnable.

Parmi les méthodes proposées pour surmonter ce défi, les méthodes multi-niveaux semblent être un candidat idéal. En réduisant de manière systématique la dimension du problème, le coût computationnel nécessaire à sa résolution peut diminuer drastiquement. Ce type d'approche est classique pour la résolution numérique des équations aux dérivées partielles (EDP), avec des garanties théoriques et des démonstrations pratiques pour expliquer leur succès.

Cependant, les méthodes actuelles d'optimisation multi-niveaux n'ont pas les mêmes garanties, ni les mêmes performances. Dans cette thèse, nous proposons de combler une partie de cet écart en introduisant un nouvel algorithme multi-niveau, IML FISTA, possédant les garanties de convergence théoriques optimales pour les problèmes d'optimisation convexes non-lisses, i.e., convergence vers un minimiseur et taux de convergence de la fonction objectif vers une valeur minimale. IML FISTA est aussi en mesure de traiter les régularisations de l'état-de-l'art en restauration d'images.

En comparant IML FISTA aux algorithmes standards sur un grand nombre de problèmes de restauration d'images: défloutage, débruitage, reconstruction de pixels manquants pour des images en couleur et des images hyperspectrales, ainsi qu'en reconstruction d'images radio-interférométriques, nous montrons qu'IML FISTA est capable d'accélérer la résolution de ces problèmes de manière significative. Le cadre d'IML FISTA est suffisamment général pour s'adapter à de nombreux autres problèmes de restauration d'images.

Nous concluons cette thèse en proposant un nouveau point de vue sur les algorithmes multi-niveaux, en démontrant leur équivalence, dans certains cas, avec les algorithmes de descente par coordonnées qui sont nettement plus étudiés dans la littérature de l'optimisation non-lisse. Ce nouveau cadre théorique nous permet d'analyser les algorithmes multi-niveaux de manière plus rigoureuse, et notamment d'étendre leurs garanties de convergence à des problèmes non-lisses et non-convexes. Ce cadre est moins général que celui d'IML FISTA, mais il ouvre la voie à une conception plus solide sur le plan théorique des algorithmes multi-niveaux.

Notations

$\mathbb{N}$	Set of *natural numbers*
$\mathbb{R}$	Set of *real numbers*
$\mathcal{H}, \mathcal{G}$	Hilbert space
$\Gamma_0(\mathcal{H})$	Set of proper, lower semi-continuous, convex functions
P	Probability
x_h	Variable living in a *fine* space
x_H	Variable living in a *coarse* space
x^i	i-th component or i-th pixel of the variable x
∇f	Gradient of the function f (uniquely valued)
$\nabla_\ell f$	Gradient of the function f with respect to the variable indexed by ℓ
$\nabla_x f$	Gradient of the function f with respect to the variable x
$\nabla^2 f$	Hessian of the function f
∂f	Subdifferential of the function f (set-valued)
prox_f	Proximal operator of the function f
ι_C	Indicator function of the set C
crit f	Set of *critical* points of the function f
$\widehat{x}$	Minimizer of a functional
$\widehat{f}$	Minimum value of a functional f
I_h^H	Restriction operator (from fine to coarse level)
I_H^h	Prolongation operator (from coarse to fine level)
$\otimes$	Kronecker product
σ_n	Noise standard deviation
σ_{PSF}	Blur standard deviation
SNR	Signal-to-noise ratio
Π_V	Projection operator onto the space V
TV	Total Variation
NLTV	Non-Local Total Variation

Part I

Introduction

Chapter 1

Introduction

> For nothing ought to be posited without a reason given, unless it is self-evident [...] or known by experience.
>
> *Guillaume d'Ockham*

1.1 Context of this thesis: inverse problems and image reconstruction

The field of "inverse problems" refers to the reconstruction of missing information from partial or degraded observations, by opposition to "direct problems" where one infers the observations from the knowledge of the parameters of the direct model. In the context of image restoration[1], the missing information is the original image, but can also include parameters describing the so-called degradation model, such as the noise level.

A famous example of inverse problems, and probably one of the oldest, is Le Verrier's discovery of the planet Neptune in 1846, by observing that the movement of Uranus did not match the prediction obtained when only taking into account the gravitational pull of Jupiter and Saturn. Le Verrier inferred the existence of another planet, Neptune, whose gravitational pull would explain the discrepancy. He presented his results to the "Académie des Sciences" on August 31, 1846, and Neptune was observed for the first time on September 23, 1846, by Johann Galle based on Le Verrier's predictions.

The study of inverse problems in a formal setting appeared at the beginning of the 20th century, and is first discussed as such by Tikhonov in 1943 [1]. Application of this setting in an imaging context followed in the 70s [2] and have ever since been a central research question.

Image reconstruction problems are a particular instance of inverse problems. As soon as an image is recorded by an instrument, be it a camera or a telescope, the resulting picture will be blurry and noisy [3]. This is an inevitable consequence that we have to deal

[1] We will alternate between image restoration and image reconstruction to qualify the same type of problems.

with. However, in many cases we understand pretty well how instruments capture and, through this process, degrade images. Naturally it raises the question of how to remove this degradation, i.e., to invert this process.

To some extent, the degradation induces a loss of information, and thus no reconstruction can be perfect. This has spurred the development of methods to best mitigate the effects of this loss, first by crafting representations of what images should look like in general [4, 5], and then by constructing algorithms able to take advantage of the knowledge of the degradation, and of these representations [2,6], to restore the image. Developments in both fields are still going on, as we do not understand completely what constitutes natural images (the term natural is commonly used to refer to what an image should look like) and what are the best algorithms in new contexts. This thesis is concerned by the latter direction: what are effective strategies to design efficient algorithms suited for the restoration and/or reconstruction of high-dimensional images.

1.2 Challenges in optimization: convergence and scalability

The methods to solve inverse problems are often based on optimization algorithms whose goal is to minimize (or maximize but these two are equivalent) an objective function. Such function is constructed from the inverse problem at hand. The common construction is to sum two different terms. The first one will control how close one image is to the observation, given the known degradation: we take an image, degrade it, and then compare it to the observed image. This term is referred to as the *data fidelity* term. It will ensure that the reconstruction matches the observation. The second term will control how close the image is to what we think the original image should look like (i.e., how natural this image is). This term is referred to as the *regularization* term. It will include *a priori* information on the reconstruction.

The solution of the optimization problem, defined by the sum of these two terms, should therefore be a trade-off between fidelity to the data, and consistency with the prior information. Reaching a trade-off, which leads to a satisfying reconstruction, is our goal when solving this problem. Therefore, choosing an optimization algorithm able to find this solution is crucial.

Hence, one of the most important questions when designing an optimization algorithm is if it can guarantee that the produced solution is the *optimal solution* of the problem, i.e., will it reach a minimizer of the problem.

Another important question is the computational cost of the algorithm. The dimension of the optimization problems considered nowadays is often very high, from millions (in typical color[2] imaging problems) to billions of variables (in real-world hyperspectral[3] imaging problems). This creates a huge computational bottleneck, on top of the storage question: each iteration of an optimization algorithm is costly, and thus we want to reduce the number of iterations to reach an optimal solution as much as possible, i.e., increase the convergence speed of the algorithm.

[2]Images with red, green and blue pixels.
[3]Images with hundreds to thousands of spectral bands.

Multilevel approaches provide ways to reduce the computational burden to reach a solution of our problem faster by modifying some iterations. This greatly improves the convergence speed of algorithms in practice.

1.3 Multilevel approaches: an intuition

To describe properly the motivations behind multilevel optimization, I prefer to begin with an analogy[4] than a technical argument. I expect that a reader having worked on optimization long enough has thought about it in similar terms at least once.

Imagine yourself blindfolded at the top of mountain. You want to reach the bottom of the valley as fast as possible. By exploring your neighborhood you can infer the slope of the mountain and take a direction of descent. Some directions are better than others, and you can find what we call the steepest descent direction (i.e., gradient descent), that will maximize your descent speed. One step at a time. However, each one of your steps can only go so far. You still have to explore your neighborhood to infer the steepest descent direction.

A classic solution to this slow speed is the momentum which could be compared to starting running in the steepest descent direction and letting your inertia guides you along the slope. But you are still blindfolded! You might go up[5].

These two analogies more or less describe the most used optimization algorithms: gradient descent and accelerated gradient descent. These two work with precise knowledge of the local landscape of our mountain (i.e., the function to minimize). However, it is straightforward to come to the conclusion that if you could remove your blindfold you would be much faster. Obviously, if that was possible, someone would have come up with an algorithm to do that already[6].

This is where multilevel optimization can come into play. To go along the analogy, it would be equivalent to removing the blindfold but still being short-sighted or myopic (not quite the optimal situation but an improvement nonetheless).

You do not need to know each rock, each patch of grass, to infer a descent direction, a rough knowledge of the slope of the mountain is sometimes sufficient. Thus, you can then take bigger steps and reach the bottom of the valley faster.

In essence multilevel optimization is the crafting of a rough knowledge of the landscape of the function to minimize, to accelerate the convergence of the underlying optimization algorithm. As we will see in this manuscript, for standard optimization problems, as long as there is some kind of structure on the function to minimize, one can derive this rough knowledge and exploit it.

There is sadly no free lunch in optimization, and building and using this rough knowledge, to accelerate the optimization, comes with a cost. Thus, a trade-off exists. Not every problem should be tackled with a multilevel algorithm; and every problem that should, cannot be tackled without careful construction.

The goal of this thesis is to provide some guidelines on the construction of multilevel algorithms for non-smooth optimization. With these insights, we propose a new multilevel

[4] I wanted to say that it is a good one to explain multilevel algorithm, but I leave this decision to the reader.

[5] Optimization algorithms rarely make the sequence fall.

[6] In fact, in some context, one can prove that such algorithm does not exist [7].

algorithm, Inexact Multilevel Fast Iterative Soft Thresholding Algorithm (IML FISTA), with state-of-the-art convergence guarantees, and we show its efficiency on a wide range of imaging problems, from the reconstruction of color images to the reconstruction of hyperspectral images. Several questions arised from the theoretical and practical study of this algorithm, we present at the end of this manuscript a new perspective on multilevel algorithm, with the point of view of block-coordinate descent algorithms, that helped answer some of these questions.

Among our experiments, provided in this thesis, we develop an instance of IML FISTA that can be applied to large-scale imaging problems in radio-interferometry. To better illustrate the potential of our proposed algorithm, we present now a summary of our contribution to this imaging problem.

1.4 An application of multilevel optimization: radio-interferometric imaging

The effort to understand how galaxies, stars, exoplanets, and the universe, formed has driven the development of new imaging methods and more computation-intensive techniques to handle the volume of data generated.

Scaling challenge in astronomy. Every day, astronomical instruments collect a huge amount of data that needs to be processed. In the optical domain, the recently launched James Webb Space Telescope[7] (JWST) produces tens to hundreds gigabytes of data per day [8] compared to the 1 or 2 of Hubble[8]. In the radio domain, the Square Kilometer Array (SKA), when delivered, is expected to produce five terabytes per second of data [9, 10]. Both fields give precious and complementary information on astronomical objects (see Figure 1.1). This calls for the development of scalable optimization algorithms with robust convergence guarantees. Multilevel algorithms are one of many solutions.

In this section we propose to illustrate the effectiveness of the method we proposed in this thesis, and where intuitions of the preceding section can lead us. To do so we present an imaging problem tackled in this thesis, which is the reconstruction of images from data obtained by radio-interferometry [11]. The next paragraphs constitute an overview of what we did on this problem and an in-depth discussion is deferred to Chapter 6.

Radio-astronomy. Complementary to optical astronomy, radio-astronomy is the field of astronomy studying objects in the radio-frequency domain, by collecting information (radio waves) through multiple antennas. Radio-interferometry is a technique used in radio-astronomy to combine the information that the antennas collect to obtain images of the sky with high sensitivity and high resolution, which would be unachievable by single antennas.

Astronomers managed since the 1950s to leverage interferometry techniques to overcome the diffraction limitation. Interferometry already had a rich history at this point (see [11]), and it led to the development of radio-interferometers: arrays consisting of multiple antennas of small diameter D spread over a large area and behaving as one

[7] https://webbtelescope.org, https://science.nasa.gov/mission/webb/
[8] https://spectrum.ieee.org/james-webb-telescope-communications

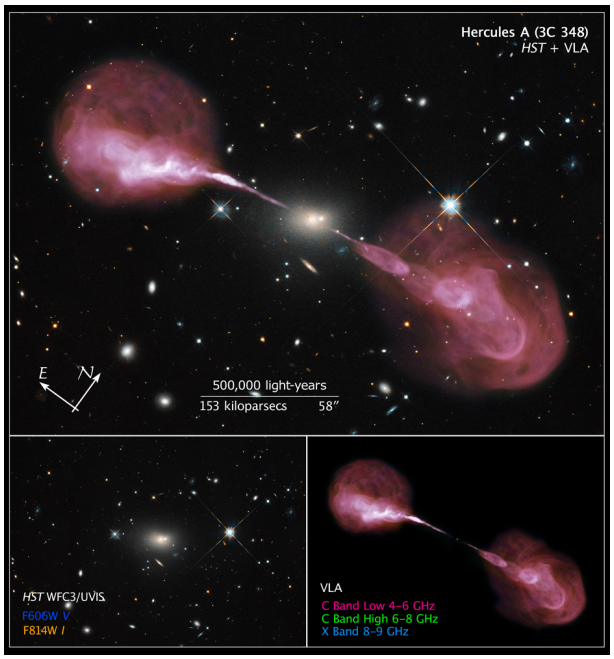

Figure 1.1: A comparison between an optical image (bottom left) and a radio image (bottom right) of the same region of the sky: the Galaxy Hercules A (3C48). Both images are combined on the top. The image in visible wavelengths was obtained by the Hubble Space Telescope, while the image in radio-wavelengths was obtained by the Karl G. Jansky Very Large Array (VLA). Credits: NASA, ESA, S. Baum and C. O'Dea (RIT), R. Perley and W. Cotton (NRAO/AUI/NSF), and the Hubble Heritage Team (STScI/AURA)

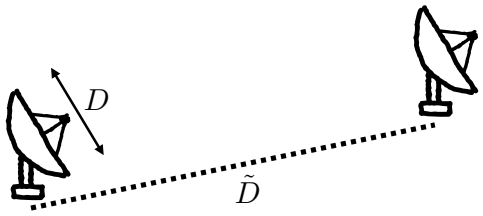

Figure 1.2: (Left) The MeerKAT radio-interferometric array in South Africa. It consists of 64 antennas, and will be a part of the future SKA. (Right) A schematic representation of a radio-interferometric array.

single antenna whose apparent diameter $\tilde{D}$ would be the largest distance between two antennas. This theoretically allows to achieve the resolution associated to a large antenna of diameter $\tilde{D}$ (see Figure 1.2).

The distance between two antennas is called a baseline. In practice, astronomers combine multiple pairs of antennas to obtain multiple baselines and thus to be able to probe the sky in multiple configurations. Measurements obtained in this way by radio-interferometers are called *complex visibilities* and unevenly cover the Fourier space. Such technique only allows to probe the sky in a sparse manner, thus requiring the use of image reconstruction techniques to achieve this resolution.

One of our contribution, in this thesis, was the development of a multilevel algorithm tailored for radio-interferometric imaging.

Multilevel optimization for radio-interferometric imaging

The number of visibilities in a radio-interferometric imaging problem is the main bottleneck for the optimization algorithm. More data means more visibilities, and thus higher computational cost. Our proposed multilevel algorithm can reduce this cost by constructing coarse approximation of the function to minimize.

To follow up with our intuition, we do not need all the visibilities to assess whether our reconstruction goes in the right direction. A natural idea is therefore to design a rough knowledge of the objective function to minimize by taking into account less visibilities. We select a subset of the visibilities to form a coarse model of the problem. In this example, we take the closest visibilities to the center of the Fourier plane, where most of the signal's energy is concentrated. Components at the center of the Fourier plane are low frequency components, and gives us this approximative knowledge of the function to minimize. Higher frequency components, which are the farthest from the center, are our small rocks and patches of grass. Hence, they may be ignored from time to time.

With such coarse model we are able to accelerate the convergence of the optimization algorithm to the solution of the problem. We obtain good reconstructions of the image in far less computation time: *IML FISTA is* 3 *to* 5 *times faster than the state-of-the-art algorithms* (see Figure 1.3 and later Chapter 6).

1.5 Summary of contributions

The main contributions of this thesis may be divided into three parts: the definition of a general multilevel framework with state-of-the-art convergence guarantees, then its application to many image restoration and reconstruction problems, including radio-interferometry; and a more theoretical part, where, equipped with a good grasp of multilevel optimization, we revisit the construction of multilevel algorithms with a new block-coordinate descent perspective.

Journal papers

1. G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves. **IML FISTA: A Multilevel Framework for Inexact and Inertial Forward-Backward. Application**

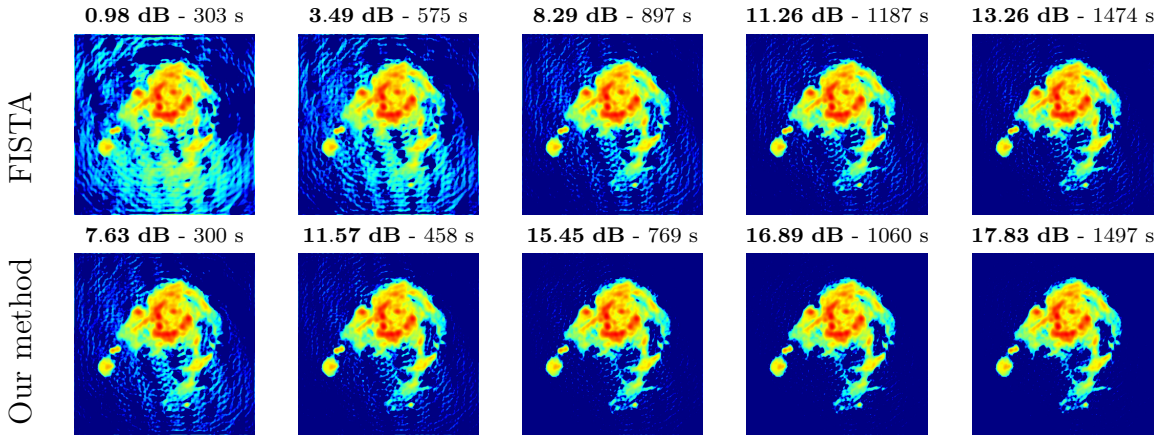

Figure 1.3: Reconstruction in log scale of a region of the M31 galaxy by FISTA (top row) and our method (bottom row) at equivalent CPU times. The legend on top of each thumbnail reads as follows: **log SNR** in dB - CPU time in seconds. Log SNR = $\mathrm{SNR}(\log_{10}(10^3 x + 1)/3, \log_{10}(10^3 x_{\mathrm{truth}} + 1)/3)$.

to Image Restoration, in SIAM Journal on Imaging Sciences, vol. 17, no. 3, pp. 1347–1376, 2024.

Conference papers

1. G. Lauga, A. Repetti, E. Riccietti, N. Pustelnik, P. Gonçalves, and Y. Wiaux. **A multilevel framework for accelerating uSARA in radio-interferometric imaging**, EUSIPCO 2024, Lyon, France.

2. G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves. **Méthodes multiniveaux pour la restauration d'images hyperspectrales**, GRETSI 2023, Grenoble, France. *Multilevel methods for hyperspectral image restoration.*

3. G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves. **Multilevel FISTA for image restoration**, ICASSP 2023, Rhodes, Greece.

4. G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves. **Méthodes proximales multiniveaux pour la restauration d'images**, GRETSI 2022, Nancy, France. *Proximal multilevel methods for image restoration.*

In preparation

1. G. Lauga, L. Briceño-Arias, E. Riccietti, P. Gonçalves, and N. Pustelnik. **Parallel, hierarchical and unbalanced block coordinate descent: convergence and a new look at multilevel optimization**, 2024 (in preparation).

1.6 Organization of the manuscript

This thesis is organized as follows. In Chapter 2 we present basic notions of convex optimization and the standard first order optimization methods. We then present in Chapter 3 the multilevel framework, and some reasons for its success in PDEs. We also present a comprehensive review of the multilevel optimization literature, with an emphasis on image restoration problems.

In Chapter 4 we present the extension of the multilevel framework to the non-smooth case and what constitutes the first major contribution of this thesis: IML FISTA. The principles underlying the algorithm are presented: extrapolation steps and estimation of the proximity operator at fine level; construction of coarse models and information transfer operators, first order coherence for non-smooth functions and decrease guarantees for multilevel steps. We also derive the convergence of our algorithm, and present a way to extend this convergence analysis to other types of multilevel first order optimization algorithms.

Then, in Chapter 5, we present several applications of IML FISTA to image restoration and reconstruction problems. We start by a benchmark to identify and select hyperparameters of the algorithm on a toy problem. We present the results of our method on image deblurring, image inpainting, and hyperspectral image restoration. Image deblurring and image inpainting are standard problems to assess the potential of optimization algorithm in imaging applications. By considering hyperspectral image restoration, we place ourselves in a high dimensional setting to assess the scalability of our algorithm.

We continue with an application of IML FISTA to a radio-interferometric imaging problem in Chapter 6. For this problem we introduce a new way of defining a multilevel algorithm by reducing the dimension of the problem along the observation instead of the image itself. This allows us to demonstrate that multilevel algorithm can be efficient in more realistic settings.

In Chapter 7 we investigate the theoretical foundations of multilevel algorithms for non-smooth optimization. The convergence guarantees associated IML FISTA, due to its generality, are not completely on the level of its great practical performance. We thus propose a construction of multilevel algorithms based on the block-coordinate descent point of view. To do so, we propose a new proximal gradient block-coordinate descent algorithm. This algorithm allows us to interpret the multilevel algorithm as a special case of a block-coordinate descent algorithm for a specific optimization problem. We present some numerical experiments to validate this point of view, and discuss the richness of this framework and its potential applications.

Chapter 2

Optimization for inverse problems

This chapter constitutes an overview of the technical background that will be used in the manuscript. We intend to present the inverse problem framework and its solving via minimization problems. Some notions about likelihood maximization, regularization and image quality metrics will be discussed. Then, by skimming through common knowledge about convex, smooth, and non-smooth optimization, we will present all the basic tools of optimization required when considering image restoration problems.

2.1 Inverse problems: optimization formulation

Many problems in image restoration can be formulated as inverse problems, whose goal is to recover the original image (or at least an image close to it) from an observation, and the knowledge of the acquisition process.

Notations. Throughout this manuscript, $\mathcal{H}$ or $\mathcal{G}$ will refer to a finite dimensional (unless stated otherwise) Hilbert space endowed with the scalar product $\langle \cdot, \cdot \rangle$, and its associated norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$.

Direct model. Formally, let z a degraded image in $\mathcal{G}$, A a bounded linear operator mapping from $\mathcal{H}$ to $\mathcal{G}$, ϵ some additive noise, and $\bar{x}$ in $\mathcal{H}$. Knowing that

$$z = \mathrm{A}\bar{x} + \epsilon, \tag{2.1}$$

we want to find $\widehat{x}$ as close as possible to $\bar{x}$. Equation (2.1) is commonly referred to as the direct or forward model.

In the following we will refer to imaging inverse problems, image reconstruction, or image restoration interchangeably as they all refer to the problem we just defined.

2.1.1 An example of image restoration

Degradation model. In this manuscript, we will focus on image reconstruction problems where the degradation operator A is linear and known. This remains fairly general as in a lot of applications the degradation can be modelled this way[1] [3, 11, 12].

[1] See Chapters 5, 6.

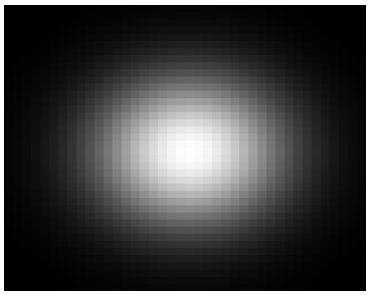

Figure 2.1: Example of a Gaussian blur's PSF.

The most common degradation we can think of in image reconstruction is the effect of a blur, i.e., a convolution of our true image with a blurring kernel [3]. To keep the setting as simple as possible, we will assume that this kernel is shift-invariant: the same blur effect will occur on every pixel. This blurring kernel is called a point spread function (PSF). It will model how the intensity of a single pixel in the image is affected by the intensity of its neighboring pixels (see Figure 2.1).

From this blur kernel, we can define the operator A as a circulant matrix that will apply the convolution to the image: each row of A computes the convolution of the image with the PSF at a given pixel. Then, we obtain a blurred image. This image is then corrupted by noise, often assumed to be Gaussian with zero mean and variance σ^2_{noise}. This assumption is quite strong. For instance, in astronomy, one is often confronted with Poisson noise, as the number of photons received by the sensor is Poisson distributed, i.e., during any given time interval the probability of receiving a certain number of photons follows a Poisson distribution [13, 14]. The subsequent noise is not additive and thus present a more challenging problem [15–17]. Fortunately, if the number of photons is large, the Poisson distribution can be approximated by a Gaussian distribution [5, 14].

The Gaussian assumption remains thus a good approximation in a lot of practical cases, and it also simplifies greatly the optimization problem as the associated functional possesses desirable smoothness properties.

Ill-posedness of the problem. It is typical of inverse problems to be severely ill-posed, which means that even if the inverse of A is available explicitly, simply computing

$$\widehat{x} = \mathrm{A}^{-1} z \tag{2.2}$$

yields a poor approximation of the solution. This estimate is highly sensitive to both blurring and noise as can be seen in Figure 2.2. Even small degradation leads to poor reconstruction.

Finding a good reconstruction of the original image may be done by maximizing the likelihood of the observation z given an image x [18]. This Bayesian interpretation is quite common, and will help us obtain the formulation of our image reconstruction problem as an optimization problem.

Bayesian formulation of the image reconstruction problem. In the Bayesian framework we aim to maximize the *a posteriori* distribution of the original image knowing

Figure 2.2: Illustration of the ill-posedness of the inverse problem on a deep field view of the galaxy cluster SMACS 0723 taken by the James Webb Space Telescope (the first image captured by this telescope). Left: original image. Middle: blurred and noisy observation. Right: reconstruction using the inverse of the operator. The reconstruction looks even more degraded than the observation. Size of the image: 2048×2048 pixels; size of the blur: 40×40 pixels; standard deviation of the noise: 0.05. Credits for the original image: NASA, ESA, CSA, STScI. Original image available here[2].

z which can be expressed with the Bayes theorem [18, 19]:

$$\mathsf{P}(x|z) = \frac{\mathsf{P}(z|x)\mathsf{P}(x)}{\mathsf{P}(z)} \tag{2.3}$$

as a function of the likelihood $\mathsf{P}(z|x)$, the marginal distribution $\mathsf{P}(z)$, and $\mathsf{P}(x)$ the *a priori* distribution for x, that describe original images following $\mathsf{P}(x) \sim \exp(-\mathsf{p}(x))$ [5, 20].

As the statistic of $z|x$ is the same as the statistic of the noise $\epsilon = z - \mathrm{A}x$, we have under the assumption that this noise is Gaussian that:

$$\mathsf{P}(z|x) = \exp\left(-\frac{1}{2\sigma_{\text{noise}}^2}\|\mathrm{A}x - z\|^2\right). \tag{2.4}$$

Thus, we deduce:

$$\mathsf{P}(x|z) = \exp\left(-\frac{1}{2\sigma_{\text{noise}}^2}\|\mathrm{A}x - z\|^2 - \mathsf{p}(x)\right)/\mathsf{P}(z) \tag{2.5}$$

The Maximum A Posteriori (MAP) approach consist in finding an image x maximizing $\mathsf{P}(x|z)$ or equivalently minimizing the following (by taking the negative logarithm of $\mathsf{P}(x|z)$):

$$\min_x \left(\frac{1}{2\sigma_{\text{noise}}^2}\|\mathrm{A}x - z\|^2 + \mathsf{p}(x)\right) \tag{2.6}$$

as the $\mathsf{P}(z)$ term does not influence the maximum/minimum value of the function. $\mathsf{p}(x)$ will encode the limited knowledge about the image we are trying to recover.

In general, changing the statistic of the noise will lead to a different expression of the probability $\mathsf{P}(x|z)$, and thus to a different optimization problem. For instance, if we assume that the noise is not Gaussian but follows a Laplace distribution, we will have an ℓ_1-norm instead of an ℓ_2-norm in what is called the data fidelity term.

[2] https://webbtelescope.org/contents/media/images/2022/035/01G7DCWB7137MYJ05CSH1Q5Z1Z.

Its formulation as an optimization problem. According to Bayesian interpretation, the image restoration problem can be generally formulated as the minimization of the sum of two functions. Following Equation (2.6), it results in the sum of a data fidelity term that we will denote L (also referred to as a "loss") and a regularization term denoted R. We then seek to find:

$$\widehat{x} \in \operatorname*{Argmin}_{x \in \mathcal{H}} F(x) := L(x) + R(x) \tag{2.7}$$

The data fidelity term controls how well an image x matches the observation and the acquisition model while the regularization term controls the knowledge we assume about the image to restore. While the Bayesian interpretation guides us for the data fidelity term choice given the noise statistics, the regularization term is more complex to define, and we will present some common choices in the following section.

2.1.2 Regularization

Sparsity inducing regularization in this manuscript. It has been remarked that a lot of signals may be represented efficiently, i.e., with a few non-zero coefficients in well-chosen bases [13,21,22]. This theory has first been extensively studied considering wavelet bases [22]. As locally regular, or locally smooth signals admit sparse representation in wavelet bases [4], seeking sparsity of the wavelet coefficients of an image provides a good denoising algorithm [22]. Simultaneously was developed the theory of *compressive sensing*, formalized in [23] and [21] to design and study representations of an image as a sparse combination of elements in a given basis. From the optimization point of view it consists in recovering the sparsest representation of the image from the observation by typically using an ℓ_1-norm to penalize the image representation in this basis [24].

In the context of image restoration, it was first formalized in [25–27] as a minimization problem with a regularization defined as follows

$$R(x) = \lambda g(\mathrm{D}x). \tag{2.8}$$

Choosing g as ℓ_1-norm and D as a wavelet transform induces a soft thresholding of the coefficients of x in a wavelet basis [22, 26].

In the following we present the most standard bases, and their operator D, under which natural images may have a sparse representation. We only present extensively the models used in this thesis. For a more complete survey see [12].

Wavelet transform. Multiresolution analysis (MRA) is a framework used primarily in image processing and functional analysis. It is designed to analyze images at multiple levels of resolution or scales [28]. This concept is central to wavelet transform, where it allows for the decomposition of a signal into different frequency components, each analyzed with a resolution that matches its scale [28].

In particular, MRA based on wavelet transform is constructed through *scaling functions* that generate a nested sequence of approximation spaces, capturing the low-frequency components of the image; and *wavelet functions* that generate detail spaces, capturing the high-frequency components or the details of the image [4]. This representation can capture both edges (neighboring pixels having a large intensity difference) and smooth regions of the image.

To come back to the denoising application of [22], the noise induces small fluctuations in the wavelet coefficients of the image (with respect to the data). Therefore, one can assume that the contribution of the coefficients whose amplitude is under a certain threshold, is due to the noise and thus might be omitted [22]. This point motivates the use of the ℓ_1-norm and associated soft-thresholding to remove noise.

Total Variation and its by-products. Natural images exhibit simultaneously some local regularity, or local smoothness, and sharp edges [4]. A good regularization should preserve these properties during the reconstruction [5]. Wavelet transform are also used for this task, but their nature may lead to unwanted artifacts in the reconstruction. Even if Total Variation regularization may also induce artifacts, its simplicity of implementation and understanding, compared to wavelets, have made it more popular over the years.

The first proposition of such regularization was done in [13] and is known as the Rudin Osher Fatemi (ROF) model for Total Variation (TV) regularization [12, 29]. In brief, TV penalizes differences between neighboring pixels, so that only sharp edges remain at the end of the optimization process, while small differences are reduced to 0 (which enforces smoothness). There exist connections between TV denoising and Haar wavelet shrinkage [22, 30, 31], but TV (and its by-products) is generally preferred as it is more robust to noise [32].

The operator D associated with the TV computes the first order differences between the component i of x (denoted x^i) and its horizontal/vertical nearest neighbors (x^{i_c}, x^{i_r}) (lower/right in the image case).

Considering only neighboring pixels in the definition of TV neglects the fact that images contain global information. This is why the Non-Local Total Variation (NLTV) was introduced in [33]. The operator D associated with the NLTV extends TV to a larger neighborhood of the current pixel i. In words, it is the operator that computes the weighted differences between the current pixel i of an image x and a subset $\mathcal{N}_i$ of pixels that are located in a large neighborhood of the current pixel i.

One can also consider the Total Generalized Variation (TGV) [34] that extends TV by considering higher order differences. The TGV allows to control the smoothness of the image at different levels, which reduces the staircase effect of TV [34]. TGV provides close result to NLTV, but it cannot be written as in Equation (2.8).

Beyond. The several regularizations we presented so far may be seen as finding sparsity in a fixed given dictionary. As an extension of these regularizations, many works proposed to learn the dictionary [35] while simultaneously restoring the images [36] (or tackling the optimization task at hand) so that the image is sparse in a proper representation basis [37].

The ever ongoing developments of machine learning, deep learning, and other data-based methods have naturally led to the design of what are called "learned" priors. One of these ideas is to train a neural network to learn the underlying distribution $\mathsf{P}(x)$ of natural images, and to use this network as a regularizer in our optimization problem. The first one of its kind was introduced in [38] where the authors replaced the proximity operator[3] of a standard penalty by a neural network trained for denoising tasks. The success of deep learning for image reconstruction has then fostered numerous works to include them in

[3] See the later sections for its complete presentation.

the rich optimization framework consistently. For instance, Regularization by Denoising (RED) was developed to incorporate deep learning based regularization [39, 40]. It was later included in a framework providing more convergence guarantees to the solution of the underlying optimization problem [41–43]. The performance of these methods is undeniable, and it was shown on real-world applications that it can outperform classical regularization methods [44]. Nevertheless, it still lacks a clear understanding, and the guarantees that are provided by variational approaches.

2.1.3 Image quality metrics

As our applications concern the restoration of images, it is important to have metrics that measure the quality of the restoration. These metrics will guarantee that the reconstruction is evaluated objectively and consistently. Nevertheless, we will also look at the visual quality of the reconstruction, to assess the quality of images that the metrics could not capture.

The ground truth $\bar{x} \in \mathbb{R}^N$ will always be available in our experiments, so we will not discuss metrics that measure image quality without reference [45–47].

Mean Squared Error. The Mean Squared Error (MSE) is the most common metric to evaluate the quality of the restoration. It is defined as:

$$\mathrm{MSE}(x, \bar{x}) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x}_i)^2 \tag{2.9}$$

and should be as low as possible.

Signal-to-Noise Ratio. The Signal-to-Noise Ratio (SNR) is a metric that measures the quality of the restoration by comparing the energy of the original image to the energy of the difference between the original image and the restored image. It is defined as:

$$\mathrm{SNR}(x, \bar{x}) = 10 \log_{10} \left(\frac{\|x\|^2}{\|x - \bar{x}\|^2} \right) \tag{2.10}$$

As its complement, the Peak Signal-to-Noise Ratio (PSNR):

$$\mathrm{PSNR}(x, \bar{x}) = 10 \log_{10} \left(\frac{N \max^2\{\bar{x}\}}{\|x - \bar{x}\|^2} \right) \tag{2.11}$$

Both should be as high as possible. PSNR measures the quality of the reconstruction by looking at the ratio between the maximum intensity value and the mean squared error, which represent the noise degradation. It can be useful when images have high dynamic range (i.e., large differences in intensity). There exist other frequently used metrics such as the SSIM [48] that measures the quality of the restoration by comparing the luminance, contrast and structure of the original image to the restored image, but is strongly correlated to the PSNR [49] in our Gaussian noise context, and can therefore be omitted.

2.2 Convex optimization

The aim of this section is to present all the concepts in optimization needed to understand or prove the results presented in the rest of the manuscript (for a more detailed description see for instance [50–55]). In brief, we discuss the main concepts of convex, smooth and non-smooth optimization.

2.2.1 Notations and reminders on convexity

We recall the classic definition of a convex set, and the definition of a convex function.

Definition 1. ***Convex set [56, Definition A.1.1.1].*** *The set $C \subseteq \mathcal{H}$ is said to be convex if for all $x, y \in C$ and $t \in (0,1)$, $tx + (1-t)y$ is in C.*

We can define a *convex function* on C as follows:

Definition 2. ***Convex function [56, Definition B.1.1.1].*** *Let C be a nonempty convex set in $\mathcal{H}$. A function $F : C \mapsto \mathbb{R}$ is said to be convex on C if for all $x, y \in C$ and $t \in (0,1)$, we have*

$$F(tx + (1-t)y) \leq tF(x) + (1-t)F(y) \tag{2.12}$$

F is said to be strictly convex if for all $x \neq y$ in C the previous inequality is strict. F is said to be strongly convex on C if there exists $\mu > 0$ such that for all $x, y \in C$ and $t \in (0,1)$, we have

$$F(tx + (1-t)y) \leq tF(x) + (1-t)F(y) - \frac{\mu}{2}t(1-t)\|x-y\|^2 \tag{2.13}$$

The following set of functions is of particular interest in optimization, as they are the building blocks of the optimization problems we will consider in this manuscript.

Definition 3. ***Proper, lower semi-continuous and convex function.*** *A function $F : \mathcal{H} \mapsto]-\infty, +\infty]$ is said to be proper if it is not equal to $+\infty$ everywhere. We will the domain of F denote by dom F the set $\{x \in \mathcal{H} | F(x) < +\infty\}$. F is said to be lower semi-continuous (l.s.c.) if for all $x \in \mathcal{H}$, if $x_n \to x$ when n goes to infinity, then*

$$F(x) \leq \liminf_{n \to \infty} F(x_n)$$

The set of proper, lower semi-continuous and convex functions defined from $\mathcal{H}$ to $]-\infty, +\infty]$ is denoted by $\Gamma_0(\mathcal{H})$.

Finally, we will need the notion of conjugate function, which is a key concept in optimization. The conjugate of a function F is a function that facilitates the characterization of the dual optimization problem associated to the minimization of F. This notion is useful to express some optimization problem in equivalent but simpler form to minimize (e.g. total variation based denoising [57]).

Definition 4. ***Legendre-Fenchel conjugate [53, Definition 13.1, Corollary 13.38].*** *The Legendre-Fenchel conjugate of a function $F : \mathcal{H} \mapsto]-\infty, +\infty]$ is defined for all $y \in \mathcal{H}$ as:*

$$F^*(y) = \sup_{x \in \mathcal{H}} \langle x, y \rangle - F(x)$$

if F belongs to $\Gamma_0(\mathcal{H})$, then $F^* \in \Gamma_0(\mathcal{H})$ and its biconjugate F^{**} is equal to F.

2.2.2 Descent directions and optimality conditions

To design an iterative minimization algorithm, we need to define descent directions, i.e., to characterize how a function will behave if we move in a certain direction at any given point. This is done formally with the directional derivative, which controls the local variation of the function.

Definition 5. ***Directional derivative [56, Definition D.1.1.1].*** *The directional derivative of F at x in the direction $d \in \mathcal{H} := \mathbb{R}^n$ is*

$$F'(x; d) = \lim_{t \downarrow 0} \frac{F(x + td) - F(x)}{t} \tag{2.14}$$

In the case of differentiable functions, the directional derivative can be expressed in terms of the gradient:

Definition 6. ***Gradient operator [53, Remark 2.55], [56, 0.4.1].*** *If F is differentiable at x, then the gradient of F at x is the unique point such that for all $d \in \mathcal{H}$:*

$$F'(x; d) = \langle \nabla F(x), d \rangle \tag{2.15}$$

In fact, the differentiability of F at x is equivalent to the existence of this unique vector $\nabla F(x)$.

In the context of image restoration, it is common to consider non-smooth functionals (e.g. ℓ_1-norm). Hence, it is important to consider how to obtain directional derivatives when the gradient is not available. The following definition characterizes the set of vectors that are below the graph of a convex function, an interesting tool to define subgradients for potentially non-smooth functions. The subdifferential of a convex function at a point x can be characterized by the sublinearity property of the function.

The following definition of the subgradient is not unique and is referred to as Subdifferential II in [56].

Definition 7. ***Subdifferential and subgradient [56, Definition D.1.2.1].*** *Let F be a proper function from $\mathbb{R}^n$ to $]-\infty, +\infty]$.*
The subdifferential $\partial F(x)$ of a function F at x is the nonempty compact convex set of $\mathbb{R}^n$ satisfying $s \in \mathbb{R}^n$ satisfying

$$\partial F(x) := \{s \in \mathbb{R}^n | F(x) + \langle s, y - x \rangle \leq F(y) \text{ for all } y \in \mathbb{R}^n\} \tag{2.16}$$

In order to produce a sequence of decreasing functional values, an iterative algorithm constructs a sequence of direction d such that $F'(\cdot; d) < 0$ at each iteration. These directions are called descent directions.

Definition 8. ***Descent direction [51, Definition VIII.1.1.1, Theorem VIII.1.1.2] [56].***
A descent direction d at x in for a convex function F is defined by the following equivalent properties

(i) $F'(x, d) < 0$;

(ii) $\langle s, d\rangle < 0$ *for all* $s \in \partial F(x)$.

Moreover, if F is convex, having $F(x + td) - F(x) \leq 0$ for some $t > 0$ implies that $F'(x; d) \leq 0$.

With this definition, one might wonder what the optimal direction would be, i.e., what would be the direction that results in the greatest decrease of the function. This is known as the steepest descent direction. It is defined as follows for smooth functions

Definition 9. ***Steepest descent direction [51, Definition II.2.1.3].*** *The steepest descent direction for a continuously differentiable convex function F at x is a descent direction d such that*

$$\widehat{d} \in \underset{\|d\|=1}{\operatorname{Argmin}} \langle \nabla F(x), d\rangle \tag{2.17}$$

This problem has a solution because $d \mapsto \langle \nabla F(x), d\rangle$ reaches its minimum, given that it is continuous and that d belongs to a compact set. Recall also that the $\langle \nabla F(x), d\rangle$ is strictly negative if d is a descent direction, thus the steepest descent direction is the most negative descent direction.

This definition can be extended to non-smooth functions by replacing the gradient by the subdifferential:

Definition 10. ***Steepest descent direction [51, Definition VIII.1.1.4].*** *The steepest descent direction for a convex function F at x is a descent direction d such that*

$$\widehat{d} \in \underset{s\in\partial F(x), \|d\|=1}{\operatorname{Argmin}} \langle s, d\rangle \tag{2.18}$$

Again, this problem has a solution because $d \mapsto \langle s, d\rangle$ reaches its minimum, given that it is continuous and that d and s belong to a compact set.

A good iterative algorithm should find the solution of the steepest descent direction at each iteration.

Now that we have seen how to characterize descent directions, we introduce the notion characterizing the set of minimizers and/or critical points of a function, which will give a sense to the convergence of the iterative algorithms (and a stopping criterion).

Definition 11. ***First order optimality condition [51, 56].*** *Let F be in $\Gamma_0(\mathcal{H})$. Then, $\widehat{x} \in \operatorname{Argmin}_x F(x)$ a minimizer of F is equivalent to $0 \in \partial F(\widehat{x})$.*

2.3 From smooth to non-smooth optimization

By making use of the tools defined in the previous section we define classical first order algorithms to solve problem of the type (2.7). We start by describing the most standard optimization algorithm: gradient descent.

2.3.1 Smooth optimization: gradient descent

Gradient descent is probably the most used optimization algorithm in the community as it is both simple and robust. The convergence of gradient descent to a critical point of the function to minimize essentially relies on the Lipschitz smoothness of the function. This property characterizes the continuity of the gradient.

Definition 12. ***Lipschitz smoothness.*** *Let $F : \mathbb{R}^N \mapsto \mathbb{R}$ be a continuously differentiable function. We say that F is β_F-smooth if for all x, y in $\mathbb{R}^N$:*

$$\|\nabla F(x) - \nabla F(y)\| \leq \beta_F \|x - y\|. \tag{2.19}$$

The well known descent lemma is a direct consequence of the Lipschitz smoothness[4] of F. This inequality is at the heart of convergence proofs of gradient descent methods.

Lemma 1. ***Descent lemma [58, 59].*** *Let $F : \mathbb{R}^N \mapsto \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant β_F. Then for any $\beta \geq \beta_F$,*

$$F(x) \leq F(y) + \langle x - y, \nabla F(y) \rangle + \frac{\beta_F}{2}\|x - y\|^2 \text{ for every } x, y \in \mathbb{R}^N. \tag{2.20}$$

Gradient descent. The previous result tells us that iterating for $k = 0, 1, \ldots$ with the step size $0 < \tau < \frac{2}{\beta_F}$

$$x_{k+1} = x_k - \tau \nabla F(x_k), \tag{2.21}$$

will produce a decrease of the function F at each iteration. Such method requires the knowledge of the Lipschitz constant β_F of the gradient. If this quantity is unavailable, a rich literature about line search methods has been developed to find the step sizes (see for instance [55, 60]).

Gradient descent is only applicable when the function to be minimized is differentiable. However, as we have previously observed, the regularization term in image restoration lacks this differentiability. Despite this limitation, there exists a tool, the proximity

[4] When a function is called β_F-smooth or Lipschitz smooth, it refers to the Lipschitz continuity of the gradient of F.

operator, that can be viewed as an implicit form of subgradient/gradient descent[5] and can be used to minimize non-smooth function.

2.3.2 Non-smooth optimization

Non-smooth optimization includes a variety of techniques, with one of the simplest being an extension of gradient descent known as subgradient descent. In this method, a subgradient is selected at each iteration to serve as the descent direction. While such algorithms can converge to a critical point under reasonable conditions, such as diminishing step sizes, their convergence rates are often suboptimal, and the behavior of the resulting sequences may lack stability. Consequently, more advanced methods have been developed, notably proximal algorithms. The proximity operator will play a fundamental role in the multilevel algorithms we will introduce later.

Proximity operator. We will refer to proximity operator the mapping defined by the following optimization problem.

Definition 13. ***Proximity operator [52, 53].*** *Let F be a function of $\Gamma_0(\mathbb{R}^N)$. Given $x \in \mathbb{R}^N$ and $\tau > 0$, the proximity operator associated to F at x is the unique point such that*

$$\operatorname{prox}_{\tau F}(x) = \underset{u \in \mathbb{R}^N}{\arg\min} \frac{1}{2\tau}\|u - x\|^2 + F(u) \tag{2.22}$$

This operator can be seen as a generalization of the projection onto convex sets, where F is the indicator function ι_C. For a lot of functions, the mapping defined by the proximity operator is known explicitly (e.g. ℓ_1-norm) or can be estimated efficiently [62,63] (cf prox-repository).

However, when dealing with the sum of two functions, such a closed form is not available and the standard approach is to split the search for a descent direction along each function. For image restoration problems (Equation (2.7)), one of these functions is often smooth, allowing us to compute its gradient. In contrast, no specific assumptions are made about the second function, which may be non-smooth. Consequently, computing its proximity operator becomes a valuable approach. This operator splitting is named proximal gradient descent or forward-backward [27]. For the sake of clarity, we split the iterations as follows:

$$x_{k+1/2} = x_k - \tau \nabla L(x_k) \tag{2.23}$$
$$x_{k+1} = \operatorname{prox}_{\tau R}(x_{k+1/2}) \tag{2.24}$$

The forward pass (Equation (2.23)) refers to the gradient descent, while the backward pass (Equation (2.24)) refers to the proximity operator application. The behavior of such algorithm can be characterized rigorously as follows.

[5] Refer to the ordinary differential equation (ODE) interpretation of gradient and subgradient descent in [61].

Proximal gradient descent. One can characterize the decrease of the objective function after one pass of proximal gradient descent, through the optimality conditions associated with the proximity operator.

Lemma 2. ***Proximal-gradient descent lemma [64].*** *Let $L : \mathbb{R}^N \mapsto \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant β_L. Let $R : \mathbb{R}^N \mapsto \mathbb{R}$ be in $\Gamma_0(\mathcal{H})$. If*

$$y = \text{prox}_{\tau R}(x - \tau \nabla L(x)), \tag{2.25}$$

then for any $0 < \tau < \frac{2}{\beta_L}$

$$L(y) + R(y) + \left(\frac{1}{\tau} - \frac{\beta_L}{2}\right) \|x - y\|^2 \leq L(x) + R(x). \tag{2.26}$$

We present the proof of this lemma as similar techniques will be used in the rest of this manuscript.

Proof. By the first order optimality condition associated with the proximity operator, we have that

$$(\forall \xi \in \mathbb{R}^N) \quad R(y) + \frac{1}{\tau}\langle x - y | \xi - y \rangle \leq R(\xi) + \langle \nabla L(x), \xi - y \rangle \tag{2.27}$$

and in particular, choosing $\xi = x$, we have

$$R(y) + \frac{1}{\tau}\|x - y\|^2 \leq R(x) + \langle \nabla L(x), x - y \rangle. \tag{2.28}$$

Now, invoking Lemma 1, it yields for any $0 < \tau < \frac{2}{\beta_L}$

$$L(y) + R(y) + \left(\frac{1}{\tau} - \frac{\beta_L}{2}\right) \|x - y\|^2 \leq L(x) + R(x). \tag{2.29}$$

□

We have now described the descent guarantee of the two (gradient and proximal-gradient descent) main first order algorithms used in optimization. Guaranteeing descent is only the first step to prove convergence of an algorithm, and the notion of convergence can take several forms that we describe in the next section.

2.3.3 Convergence of optimization algorithms

In order to characterize and compare algorithms, we look at the convergence guarantees they provide. In this manuscript we will need three notions of convergence to assess the performance of the algorithms we designed with respect to those of the literature. We will derive them for the minimization of a continuously differentiable function F, for the sake of clarity, but all these notions remain relevant for non-smooth functions.

The first and most important notion in our context is the convergence to a minimizer of the function. In inverse problems, this guarantees that the restored image is the desired solution.

Theorem 1. ***Convergence of iterates [65].*** *Let F be a convex function. Let $(x_k)_{k\in\mathbb{N}}$ be generated by an algorithm is said to converge to a minimizer $\widehat{x}$ of F if the generated sequence of iterates is such that its limit exist and respects*

$$\lim_{k\to+\infty} x_k = \widehat{x} \in \arg\min_{x\in\mathcal{H}} F$$

Remark 1. *The convergence of the iterates to a minimizer implies the convergence of the objective function values to the minimum value. The converse is not true in general.*

Theorem 2. ***Convergence of objective function values [65].*** *Let F be a convex function. Let $(x_k)_{k\in\mathbb{N}}$ be generated by an algorithm is said to converge in function values if*

$$\lim_{k\to+\infty} F(x_k) = \min_{x\in\mathcal{H}} F(x) = \widehat{F}$$

The last notion of convergence that will be used in this manuscript is the global convergence.

Theorem 3. ***Global convergence [55, Chapter 3].*** *Let F be a continuously differentiable function. An algorithm is said to be globally convergent if the generated sequence of iterates is such that starting from any initial point*

$$\lim_{k\to+\infty} \nabla F(x_k) = 0$$

Of the three convergence guarantees we are interested in, this one qualifies as the weakest because we can find a function that is smooth and convex for which an algorithm will converge globally, but the underlying sequence will not (see Section 3.2.3).

Rate of convergence. Now that we have seen how an algorithm produces convergent sequences, the next step is to quantify the speed at which the sequences converge. A great deal of research has been dedicated to this topic and for most of the classical algorithms and classical classes of functions, we know precisely which rate of convergence one can expect from a given algorithm. For instance, one can show that the optimal convergence rate of gradient descent for a convex function with β_F-Lipschitz continuous gradient is $\mathcal{O}(1/k)$, i.e., starting from an initial guess x_0,

$$F(x_k) - F(\widehat{x}) \leq \frac{\beta}{2k}\|x_0 - \widehat{x}\|^2,$$

where $\widehat{x}$ is a minimizer of F. This is not the best rate ($o(1/k)$) of convergence one can theoretically hope for as we will see in the next sections. What is interesting to note is that in order to prove the previous rate of convergence, one has to invoke the convexity of F and the related inequalities. As an example, strong convexity would yield linear convergence rates:

$$F(x_k) - F(\widehat{x}) \leq \left(1 - \frac{\mu}{\beta_F}\right)^k \|x_0 - \widehat{x}\|^2, \tag{2.30}$$

where μ is the strong convexity constant of F.

2.4 Acceleration techniques

In this section we present some acceleration techniques used to improve the practical and/or theoretical convergence rate of first order optimization algorithms.

2.4.1 Momentum, inertia and other extrapolation steps

One of the simplest, and most efficient acceleration techniques is the addition of inertia or momentum to the gradient descent/proximal gradient descent algorithm. The two most known are referred to as Polyak's momentum or heavy ball method [66], and Nesterov's method [67] or FISTA [68]. Only the latter has been proved to be optimal, therefore we focused on this one in this manuscript.

It is well known that for problem of the form (2.7), the optimal worst case rate of convergence is $1/k^2$ [7, 67, 68]. To reach this rate of convergence, it was first proposed in [67] to add extrapolation steps to a classical gradient descent algorithm. This was later proposed for proximal gradient descent in [68] with the following iterations, starting from $x_0 = y_0 \in \mathcal{H}$ and $t_0 = 1$. For $k = 0, 1, ...$

$$x_{k+1} = \text{prox}_{\tau_k R}(y_k - \tau_k \nabla L(y_k)) \tag{2.31}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{2.32}$$

$$y_{k+1} = x_{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_{k+1} - x_k) \tag{2.33}$$

where $0 < \tau_k < 1/\beta_L$, the Lipschitz constant of the gradient of L.

A basic intuition on adding extrapolation steps is to compensate the fact that gradient steps, and as a result proximal gradient steps, converge in norm to 0 when approaching minimizers or critical points. This slows down the sequence. At the same time $\frac{t_k - 1}{t_{k+1}}$ starts from 0 and converges to 1 as k goes to infinity. While gradient steps decrease in size, extrapolation steps gain in importance thus "accelerating" the convergence speed of the sequence.

Convergent FISTA and Nesterov's rule. It is also notable that even though the optimal convergence rate of objective function values is recovered with extrapolation steps defined by Equation (2.32) and (2.33), the convergence of the sequence $(x_k)_{k\in\mathbb{N}}$ to a solution of Problem (2.7) is not known, and thus not guaranteed in a general setting.

It was later shown in [69] and [70], by using a sequence of extrapolation steps [69] so that FISTA converges to a solution of Problem (2.7), that FISTA with the "Nesterov" sequence of extrapolation steps defined as in Equation (2.32) constitutes an edge case.

This is characterized by the following inequality for all $k \in \mathbb{N}$:

$$t_k - t_{k+1}^2 + t_{k+1} > 0 \tag{2.34}$$

For sequence $(t_k)_{k\in\mathbb{N}}$ ensuring this strict inequality, convergence of the sequence $(x_k)_{k\in\mathbb{N}}$ has been obtained in the convex case to a minimizer of the Problem in [69][6] and with

[6]The convergence was obtained in a weak sense, which is equivalent to the strong convergence (cf Theorem 1) in finite dimension.

errors in [70]. The so-called Nesterov sequence is such that for all $k \in \mathbb{N}$:

$$t_k - t_{k+1}^2 + t_{k+1} = 0 \tag{2.35}$$

For the rest of this manuscript we will refer to Equations (2.34) and (2.35) as the Nesterov's rule (including thus the equality case). The proof of convergence to a minimizer relies on a Lyapunov analysis [69, 70], where an energy is to be minimized. Having the strict inequality being transformed into an inequality implies that the bound obtained on the energy would be as meaningful as showing $0 \geq 0$, i.e., in this case a Lyapunov analysis cannot tell us anything about the convergence of the sequence.

Remark 2. *Note that in FISTA (and to the best of our knowledge every other related algorithms) the sequence $(y_k)_{k\in\mathbb{N}}$ does not converge to a solution of Problem (2.7).*

Several incremental improvements of FISTA have been made since in the literature. The choice of the inertial sequence parameters is highly dependent on the geometry of the problem and can be optimally chosen under certain assumptions (e.g. strong convexity [71], [72], restart [73], automatic choice of the parameters [74]).

Moreover, one can recover convergence guarantees even in the case where $t_k - t_{k+1}^2 + t_{k+1} < 0$. Such study was conducted in [75] under the sequence framework of [69, 70], which will also be at the core of our analysis.

Through the ODE interpretation of gradient descent, and proximal gradient descent, more elaborate studies of inertial algorithms have been conducted [76–78]. They shed an interesting perspective on the dynamics of the algorithm.

2.4.2 Variable metric and preconditioning

Variable metric. The idea of variable metric methods is to adapt the metric of the space, in which the optimization is performed, to the local geometry of the function to minimize. This is done by replacing the Euclidean metric by a positive definite matrix H_k at each iteration k. The most famous variable metric method in the smooth case is the Quasi-Newton algorithm: the Broyden-Fletcher-Goldfarb-Shanno (BFGS) or LBFGS algorithm [55] where an approximation of the Hessian matrix is used at each iteration to adapt the metric. Such idea can be extended to the non-smooth case by changing the metric used in the definition of the proximity operator (Definition 13) [79–81].

Preconditioning. As the name suggests, preconditioning aims to improve the conditioning of the problem, so that it is easier to solve. For the sake of argument, consider the following optimization problem

$$\min_x F(x) = \frac{1}{2}\|\mathrm{A}x - z\|^2$$

This function has a constant of strong convexity constant of μ which is the smallest eigenvalue of $\mathrm{A}^T\mathrm{A}$ and a Lipschitz smoothness constant of β_F which is the largest eigenvalue of $\mathrm{A}^T\mathrm{A}$. This function being strongly convex, the closer to 1 the ratio between μ and β_F the smaller the convergence rate (Equation 2.30). One can improve this ratio by preconditioning the problem, i.e., reducing β_F/μ, and there exist many ways to do so [55].

2.5 Conclusion

We have presented the main tools and concepts to understand and study the behavior of first order optimization algorithms. For smooth and non-smooth optimization, gradient and proximal gradient descent are the standard. Several acceleration techniques have been proposed over the years to improve the convergence rate of these algorithms. Due to the increasing scale of optimization problems to solve, development of techniques able to alleviate the computational burden is of paramount importance. In the next chapter we will present the method studied in this manuscript: multilevel optimization.

Chapter 3

A short presentation of multilevel optimization

This chapter constitutes an overview of the technical background that will be used in the manuscript to define and study multilevel algorithms. We first present its application to solve partial differential equations (PDEs). This presentation will highlight the theoretical argument that propelled multigrid methods to be state-of-the-art when solving certain PDEs. This success has inspired the adaptation of this framework to optimization, and we will try to provide a comprehensive overview of the related research found in the literature.

By the end of this chapter, we aim to have clearly outlined the strengths and limitations of existing multilevel optimization methods, thereby motivating the focus of the work conducted in this thesis.

Multilevel or multigrid? In the rest of the manuscript, and in the current literature, the terms multilevel and multigrid are often used interchangeably. It is quite straightforward to understand the use of multigrid, but the rationale behind using multilevel is somewhat unclear[1]. From my perspective, it could be justified given that some works define "coarse" approximations of the problem without necessarily reducing the dimension, or not along the line of reducing the dimension of the variable space (e.g. the resolution of an image)[2].

3.1 Multigrid methods

Multigrid methods were initially developed to solve differential equations by utilizing a hierarchy of discretization grids [84–86]. These concepts naturally arise when dealing with problems originating from the discretization of continuous models. The idea is to vary the level of discretization to solve problems with different levels of precision (Figure 3.1). Coarse, i.e., less accurate, solutions, which are computationally cheaper, can be leveraged to accelerate the computation of more accurate, but costly, fine solutions.

[1] There is no apparent connection with bilevel optimization [82,83], but maybe it could be interesting to try to draw one!

[2] See Appendix A.1 for some references.

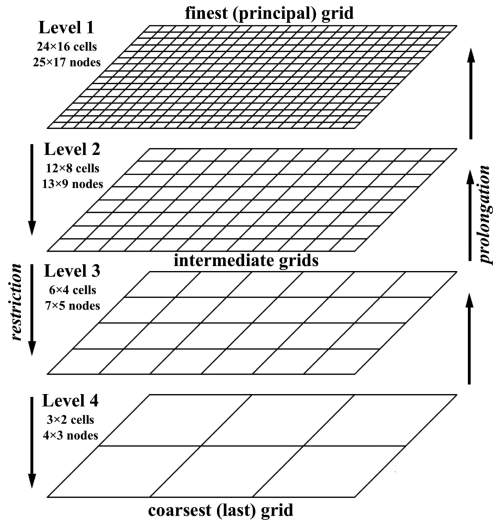

Figure 3.1: Illustration of the hierarchy of grids used in multigrid methods from [87]. The fine grid is used to solve the problem with high accuracy, while the coarse grid is used to accelerate the computation.

Multigrid methods formalize this intuition by constructing hierarchies of discretizations and utilizing all of them to solve the problem more efficiently at the desired fine resolution. These techniques have a rich history in solving differential equations, and the following paragraphs will discuss the foundational concepts that contribute to their practical and theoretical effectiveness.

3.1.1 The purpose of multigrid methods

To illustrate the motivation behind multigrid methods, we consider the following one dimensional equation with Dirichlet boundary conditions [86]:

$$-u''(x) + u(x) = f(x) \text{ in } \Omega = (0,1) \text{ with } u(0) = u(1) = 0. \tag{3.1}$$

The rest of the presentation will be based on this equation, and the principles presented here are derived exhaustively in [86, Chapter 1 and Chapter 2]. This example is quite standard in the literature and serves as an introduction in most survey/review of multigrid methods for solving partial differential equations [84–86]. In the context of this problem, one can highlight that using coarse grids is not only computationally cheaper, but also that it improves the theoretical convergence rate of the algorithm.

This presentation is quite long, but it is necessary, I think, to understand why multigrid methods are so efficient, and to understand the gap between their success in solving PDEs and their lack of success[3] (in comparison) in optimization.

To solve this problem consider a grid of evenly spaced data points $x_i = ih$ for $i = 0, 1, \ldots, N$ with $h = 1/N$ and N the number of points. This will constitute our fine grid

[3]This is not intended to be derogatory to the field, but multilevel optimization has yet to display the fast convergence of multigrid methods for PDEs. Compare for instance the convergence plots in [88, Figure 1], where a multilevel proximal algorithm is applied to a PDE's problem to plots in Chapter 5 or in [89,90], even though algorithms are conceptually similar.

with a mesh size of h. A finite difference approximation of the previous equation reads, with $u_i = u(x_i)$ and $z_i = z(x_i)$:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + u_i = z_i \text{ for } i = 1, \dots, N-1 \tag{3.2}$$

which can be concisely written as

$$\mathrm{A}u = z \tag{3.3}$$

where

$$\mathrm{A} = \frac{1}{h^2}\begin{pmatrix} 2+h^2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2+h^2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2+h^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2+h^2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2+h^2 \end{pmatrix} \tag{3.4}$$

and $u = (u_1, \dots, u_{N-1})^T$ and $z = (z_1, \dots, z_{N-1})^T$, are discrete approximations of the true solution and the data. An equation with the boundary conditions completes this linear system: $u_0 = u_N = 0$. We will omit in the following the dependence in h for the sake of clarity.

Solving a differential equation on a grid amounts thus to solving a linear system of equations of size N. Direct methods, such as Gaussian elimination, work perfectly well for simple problems such as this one, but are quite limited by the size of the system. This has spurred the developments and study of iterative methods such as the Jacobi, or Gauss-Seidel methods that aim to solve the linear system by improving the solution at each iteration, starting from an initial guess. Such methods are also referred to as relaxation methods, and have a long history in the solving of partial differential equations [91, 92] (or in general [93]). These methods are more general but can be really slow to converge, and multigrid methods are the remedy to this problem (for PDEs).

3.1.2 Solving PDEs with multigrid methods

In this section we study the behavior and convergence of the iterative methods that propelled the construction of multigrid methods.

Solving a linear system. Solving the linear system $\mathrm{A}u = z$ can be seen as a fixed point problem. The solution $\hat{u}$ verifies the following "fixed point" equation:

$$\mathrm{A}\hat{u} - z = 0 \tag{3.5}$$

An iterative method aims to find a sequence $(u^k)_{k\in\mathbb{N}}$ that converges to $\hat{u}$. We have seen in the previous chapter the standard fixed point iterative methods in optimization: gradient descent. It aims to find for a function F the solution $\hat{u}$ of:

$$\nabla F(\hat{u}) = 0 \tag{3.6}$$

We could formulate Problem (3.3) as an optimization problem and solve it with gradient descent, but here we are interested in the behavior of iterative methods only applicable

to linear systems. It is in this particular context that one can see the practical and theoretical advantages of multigrid methods. We will now detail the construction of the Jacobi and Gauss-Seidel methods, and then introduce the multigrid method.
Given a vector u we can define the algebraic error as

$$e = u - \hat{u}.$$

This error is obviously not available and is approximated by the residual r defined as:

$$r = z - \mathrm{A}u.$$

Now split A into its diagonal components D, its lower triangular part L and its upper triangular part U so that $\mathrm{A} = D - L - U$. The Jacobi method consists in iterating the following steps:

$$u^{k+1} = D^{-1}(L+U)u^k + D^{-1}z, \tag{3.7}$$

until a solution is found. The Gauss-Seidel method is a slight refinement of the Jacobi method. The update is as follows:

$$u^{k+1} = (D-L)^{-1}Uu^k + (D-L)^{-1}z. \tag{3.8}$$

As $\hat{u}$ is the solution of the linear system, it is a fixed point of the previous iterative methods, thus we can express the error at iteration k, in the case of the Jacobi method, as

$$\begin{aligned} e^{k+1} &= Re^k \\ &= R^{k+1}e^0, \end{aligned}$$

with $R = D^{-1}(L+U)$. If $\|R\| < 1$ then the error converges to 0 as k goes to infinity. Now we want to look at the rate of convergence under which the components of the error go to 0. To do so, we introduce the weighted Jacobi method that changes the error update by using $R_\omega = (1-\omega)\mathrm{Id} + \omega R$. It can be shown that the eigenvalues of R_ω and A are linked by the following relationship [86]:

$$\lambda(R_\omega) = 1 - \frac{\omega}{2}\lambda(\mathrm{A}).$$

The eigenvalues of A (given in Equation (3.4)) are given by [86]:

$$(\forall 1 \leq \ell \leq N-1), \quad \lambda_\ell(\mathrm{A}) = 4\sin^2\Big(\frac{\ell\pi}{2N}\Big),$$

which yields that:

$$(\forall 1 \leq \ell \leq N-1), \quad \lambda_\ell(R_\omega) = 1 - 2\omega\sin^2\Big(\frac{\ell\pi}{2N}\Big).$$

The corresponding eigenvectors of A are the following:

$$(\forall 1 \leq \ell \leq N-1), \quad \mathbf{v}_\ell = \left(\sin\Big(\frac{j\ell\pi}{N}\Big)\right)_{0\leq j\leq N}.$$

It allows us to express the evolution of the error in the basis formed by these eigenvectors.

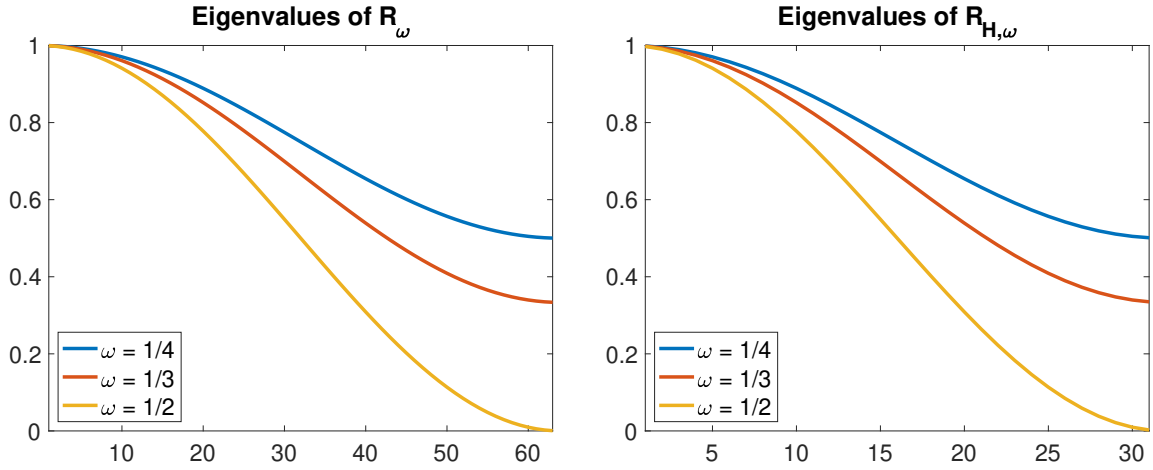

Figure 3.2: (Left) Eigenvalues ($x-$axis indexes ℓ, $y-$axis indicates the value) of the weighted Jacobi method for the one dimensional diffusion equation with $N = 64$. The higher the frequency of the mode, the lower the reduction factor, and thus the faster the convergence. (Right) Eigenvalues of the coarse (denoted by H) level weighted Jacobi method.

Remark 3. *The oscillations of the eigenvectors of* A *grow with* ℓ *while the eigenvalues of* R_ω *decrease with* ℓ*. This is the reason why the high frequency modes of the error converge faster than the low frequency modes.*

There exist $(\epsilon_\ell)_{1\leq\ell\leq N-1} \in \mathbb{R}^{N-1}$ such that

$$e^0 = \sum_{\ell=1}^{N-1} \epsilon_\ell \mathbf{v}_\ell.$$

After k iterations, the error reads

$$e^k = \sum_{\ell=1}^{N-1} \epsilon_\ell \lambda_\ell^k(R_\omega)\mathbf{v}_\ell,$$

as eigenvectors of A are eigenvectors of R_ω. From this expression one can deduce that the $\ell-$th component of the error has been reduced by a factor $\lambda_\ell^k(R_\omega)$ after k iterations. One also can see in Figure 3.2(Left) that the low frequency modes of the error have the highest reduction factor, i.e., the closest to one (and thus the slowest convergence) while the high frequency modes have the lowest reduction factor (and thus the fastest convergence). The method smooths the error quite efficiently, but its low frequency modes remain untouched, as they appear smooth on the grid.

This is where the coarse grid comes into play. This coarse grid contains half the number of points of the fine grid, spaced by $2h$. With an approximation of no consequence, we can say that the coarse grid iterative equation will be the same as the fine grid iterative equation, with a smaller number of points and a factor 2 in front of each h. Therefore, we can reuse the study of the eigenvalues of A to study the eigenvalues of the coarse model operator, that we denote A_H. The eigenvalues of the coarse weighted Jacobi method follow a similar trend as those of A that is depicted in Figure 3.2(Right).

In practice A_H is constructed using a Galerkin approximation [86, 94], that is the restriction/projection[4] of A to the coarse grid. The restriction is done using a linear operator I_h^H (see Definition 14 and Equation (3.12) below) that sends information from the fine grid to the coarse grid. The prolongation operator I_H^h sends information from the coarse grid to the fine grid. The Galerkin approximation reads:

$$\mathrm{A}_H = I_H^h \mathrm{A} I_h^H \tag{3.9}$$

Nevertheless, the eigenvalues behave similarly as in Figure 3.2.

By construction, as the error is projected to this coarse grid, the modes from around $N/4$ to $N/2$ on the finest grid will become high frequency modes on the coarse grid. Thus, they will enjoy the fast convergence associated with the iterative method. Coarsening the grid will result in targeting lower and lower frequency modes.

Most of the relaxation/iterative schemes have this smoothing property, and are thus well suited to be accelerated by multigrid methods [86]. Moreover, for multigrid methods to be fully valuable, the residual of the problem should be smooth before being sent to the coarse grid, so that it contains only low frequency components. Relaxation steps and multigrid steps complement each other nicely in the solving of differential equations. The common steps of multigrid methods are the following [84–86, 95]:

- Smoothing: apply relaxation steps to the current iterate;
- Restriction: project the resultant residual to the coarse grid;
- Smoothing on the coarse grid: apply relaxation steps to the projected residual;
- Prolongation: send the smoothed residual from the coarse grid to the fine grid;
- Correction: correct the current iterate with the smoothed residual.
- Smoothing: apply relaxation steps to the corrected iterate.

This procedure is repeated until convergence.

3.2 Multilevel optimization

What can be considered the first extension of the multilevel/multigrid framework to optimization was done in the seminal paper of S.G. Nash [96]. These methods are usually referred to as multilevel methods even though some also use multigrid (e.g. [88]). Remarking that, just as linear systems arising from PDEs problems, optimization problems can often be seen as discretization of problems in infinite[5] dimensional spaces (e.g. variational approaches), it was more than natural to use a hierarchy of such discretizations to tackle the optimization problem in a high dimensional space of interest. An idea straightforwardly inherited from the multigrid literature on PDEs solving.

We note however that an application of the multigrid framework to optimization had been done previously in [97] where the authors used an algebraic multigrid approach to

[4] The two terms refers to the action of sending fine information to the coarse level.

[5] For instance, the underlying objects in an image do not have finite resolution.

compute a descent direction for Newton's method by solving with it the following linear system for a twice continuously differentiable function F :

$$\nabla^2 F(x)d = -\nabla F(x). \tag{3.10}$$

$\nabla^2 F(x)$ (resp. $\nabla F(x)$), discretized on a grid would be A (resp. z) in the previous section. A discretized d is then computed using a multigrid method and then used as a descent direction to solve a nonlinear optimization problem.

In its article [96], Nash described MG/OPT as a general framework to solve optimization problems. This framework is the basis of most of what is qualified as multilevel optimization today. The iteration scheme follows the one of classical multigrid scheme:

- N_0 gradient or optimization steps on the fine level problem (equivalent to smoothing steps);
- Projection of the current iterate and gradient value to the coarse level;
- Minimization of the coarse model corrected by the projected gradient value;
- Prolongation of the result to the fine level;
- Correction of the current iterate using a line search to guarantee descent inspired from the trust region literature [55, 98];
- N_1 gradient steps on the fine level problem.

Nash proved the global convergence of his algorithm using the following argument [96, Theorem 1]. MG/OPT combines gradient descent – a method known to be globally convergent when paired with an appropriate line search [55] – with multigrid updates that ensure the objective function does not increase. Since gradient descent is globally convergent, MG/OPT inherits this property as well. However, this guarantee is relatively weak. As we will discuss in Section 3.2.3, it is possible to construct functions and optimization algorithms that provide similar non-increasing behavior as MG/OPT without ensuring that the sequence of iterates actually converges.

Since this work, multilevel approaches have not been as successful as multigrid methods for solving PDEs in the optimization community. I think that the main reason for this lack of growth is the difficulty to obtain great practical performance – something these methods are capable of in an optimization context – while maintaining the implementation and tuning complexity of the algorithm low. We hope that the next section will highlight this clearly for the reader[6].

3.2.1 Core principles

In this section we present the core principles required to design a multilevel algorithm with some theoretical guarantees. The presentation is done for a two levels algorithm, and we will reuse this format for the rest of the manuscript. If the extension to more than two levels is not straightforward in some contexts, necessary details will be presented.

[6] I do not claim to have circumvented completely this difficulty in this thesis.

Note that even though the content of this section was already known in the literature, we worked to clarify and organize the presentation of these principles in this context.

Let us introduce a notation that will follow us through the rest of this manuscript. We index by h what we refer to as the fine level, and by H what we refer to as the coarse level. Unless stated otherwise the fine level objective function F (Equation (2.7)) will be denoted as F_h from now on.

In contrast to multigrid methods for linear system solving, we do not have a way to quantify the distance of our current iterate to a minimizer of the fine level problem (an equivalent to what would be the residual). Thus, there is an ambiguity around the definition of a good coarse level that we need to solve.

Information transfer operators. First, we need to define a coarse space where coarse variables and functions will live in. Suppose that the fine level problem is defined on $\mathbb{R}^N$. We define $N_h := N$ as the dimension of the fine level problem. The coarse level dimension will be denoted N_H. The standard assumption is $N_H < N_h$. Now to go from fine level to coarse level and *vice versa*, we have the following operators

Definition 14. ***Information transfer operators.*** *Let $I_h^H : \mathbb{R}^{N_h} \mapsto \mathbb{R}^{N_H}$ and $I_H^h : \mathbb{R}^{N_H} \mapsto \mathbb{R}^{N_h}$ be linear operators. They are called coherent information transfer operators (CIT) if there exists $\nu > 0$ such that*

$$I_H^h = \nu \left(I_h^H\right)^T \tag{3.11}$$

I_h^H is referred to as the *restriction operator* that sends information from the fine level to the coarse level, and reciprocally I_H^h is the *prolongation operator* that sends information from the coarse level back to the fine level. In imaging, to satisfy Equation (3.11), we usually construct the restriction operator first and then take its normalized transpose to define I_H^h.

There are many ways to construct such operators. The most standard CIT operator for multilevel methods is the dyadic decimated and weighted operator [86]. In the particular case of squared grids of size $\sqrt{N_h} \times \sqrt{N_h}$ and $\sqrt{N_H} \times \sqrt{N_H}$ at fine and coarse level respectively, and for $N_H = N_h/4$ corresponding to a decimation factor of 2 along rows and columns, the restriction operator reads:

$$I_h^H = \frac{1}{16} \underbrace{\begin{pmatrix} 2 & 1 & 0 & \dots & & & 0 \\ 0 & 1 & 2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & & & & 0 \\ 0 & \dots & & 0 & 1 & 2 & 1 \end{pmatrix}}_{\sqrt{N_h}/2 \times \sqrt{N_h}} \otimes \underbrace{\begin{pmatrix} 2 & 1 & 0 & \dots & & & 0 \\ 0 & 1 & 2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & & & & 0 \\ 0 & \dots & & 0 & 1 & 2 & 1 \end{pmatrix}}_{\sqrt{N_h}/2 \times \sqrt{N_h}} \in \mathbb{R}^{N_H \times N_h}. \tag{3.12}$$

$\otimes$ denotes the Kronecker product.

The pair $(I_h^H,\ I_H^h)$ provides a simple and intuitive way to transfer information back and forth between fine and coarse scales, by means of linear B-spline interpolation. Other operators of the form of (3.12) corresponding to higher order interpolation have been proposed in [99] and are commonly used in multigrid methods for solving PDEs [100].

Definition of coarse level functions. The rule of thumb to define coarse level functions is to take reduced order version of the fine level function. If the reader is interested this notion can be rigorously defined using the theory of $\Gamma-$convergence [101], but this theory is not at all necessary to create multilevel hierarchy.

Suppose that

$$F_h(\cdot) = \| \cdot - z\|^2_{\mathbb{R}^{N_h}}$$

is the fine level objective function. Then a reduced order version of F_h would naturally be

$$F_H(\cdot) = \| \cdot - I_h^H z\|^2_{\mathbb{R}^{N_H}},$$

where $I_h^H z$ is the restriction of z to the coarse space. This is not standard in multigrid methods for PDEs: often the observation z (see Section 3.1.1) is of small size compared to u, and therefore remains unchanged.

Now, one needs to define equivalently reduced order version of the linear operators involved in the optimization problem. Denote by $\mathrm{A}_h := \mathrm{A}$ the fine level degradation operator. There is an equivalence between minimizing $\|\mathrm{A}_h \cdot - z\|^2$ and solving $\mathrm{A}_h x = z$. Multigrid method to solve such linear system naturally define the coarse matrix A_H as the restriction of A_h to the coarse space:

$$\mathrm{A}_H = I_H^h \mathrm{A}_h I_h^H$$

This approximation of A_h is often referred to as the Galerkin approximation [86, 94] and was coined for the first time in multilevel optimization in [98]. In the case of multigrid methods, only the columns of A are modified, and the rows remain unchanged to match the unchanged observation z.

First order coherence between levels. Designing natural approximation of the fine level objective function is not enough to guarantee that such coarse model will help the optimization of the fine level problem. One way to ensure that it does is to impose coherence between levels. Let us define a smooth coarse model for smooth functions to minimize.

Definition 15. ***Coarse model F_H for smooth functions.*** *A continuously differentiable coarse model F_H is defined for the point $x_h \in \mathbb{R}^{N_h}$ as:*

$$F_H = L_H + R_H + \langle v_H, \cdot \rangle, \tag{3.13}$$

where

$$v_H = I_h^H \left(\nabla L_h(x_h) + \nabla R_h(x_h)\right) - \left(\nabla L_H(I_h^H x_h) + \nabla R_H(I_h^H x_h)\right). \tag{3.14}$$

Adding the linear term $\langle v_H, \cdot \rangle$ to $L_H + R_H$ allows to impose the so-called *first order coherence*.

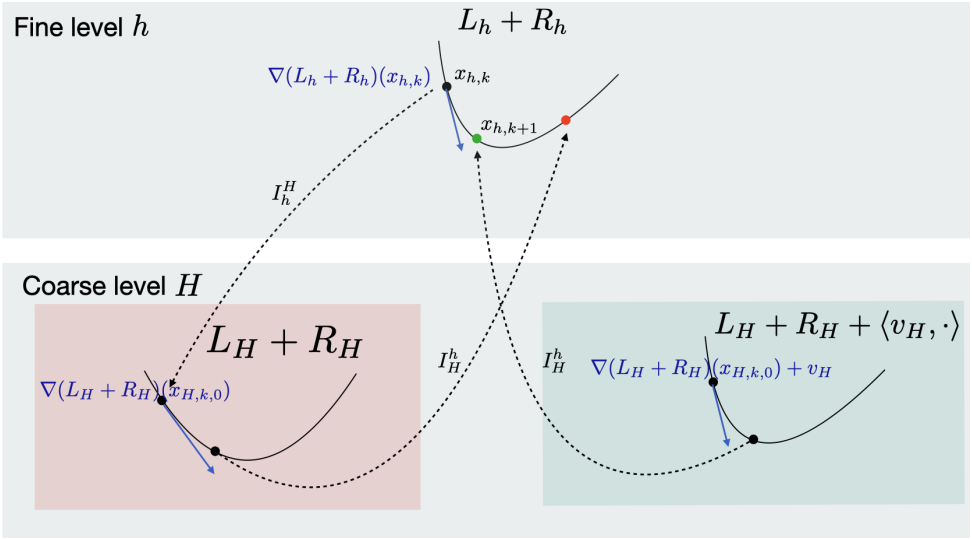

Figure 3.3: Illustration of the first order coherence between two smooth functions $L_h + R_h$ and $L_H + R_H$. Left lower part: without first order coherence, points decreasing $L_H + R_H$ do not necessarily decrease $L_h + R_h$. Right lower part: first order coherence rotates the graph of $L_H + R_H$ around $x_{H,0}$ so that decreasing $L_H + R_H$ also entails decreasing $L_h + R_h$.

Definition 16. ***First order coherence [89, 90, 96].*** *The first order coherence between the objective function F_h at the fine level and the coarse level objective function F_H is verified in a neighborhood of x_h if the following equality holds:*

$$\nabla F_H(I_h^H x_h) = I_h^H \nabla \left(L_h + R_h\right)(x_h). \tag{3.15}$$

The following lemma states that the coarse model proposed in Definition 15 verifies the first order coherence.

Lemma 3. *If F_H is given by Definition 15, it necessarily verifies the first order coherence (Definition 16).*

Proof. Considering the gradient of the coarse model F_H and combining it with the definition of v_H in Equation (3.14), yields

$$\begin{aligned} \nabla F_H(I_h^H x_h) &= \nabla L_H(I_h^H x_h) + \nabla R_H(I_h^H x_h) + v_H, \\ &= I_h^H \left(\nabla L_h(y_h) + \nabla R_h(x_h)\right). \end{aligned} \tag{3.16}$$

The first order coherence defined in Definition 16 is thus verified. □

This condition ensures that, in the neighborhood of the current iterates x_h and $I_h^H x_h = x_{H,0}$, the fine and of the coarse level objective functions are coherent up to order one. Figure 3.3 illustrates the effect of the first order coherence on the alignment of the gradients of smooth objective functions at fine and coarse levels.

Now that we have defined the coarse model, we can look at what happen when we minimize it. We apply repeatedly the following gradient steps on F_H:

$$x_{H,\ell+1} = x_{H,\ell} - \gamma_H \nabla F_H(x_{H,\ell}),$$

where $0 < \gamma_H < 2/\beta_H$. β_H is the Lipschitz constant of the gradient of F_H, which is unaffected by the first order coherence term.

We will thus assume that after m iterations of gradient descent on F_H we obtain $x_{H,m}$ that is such that $F_H(x_{H,m}) \leq F_H(x_{H,0})$. With the first order coherence, we guarantee that the term $I_H^h(x_{H,m} - x_{H,0})$ is a descent direction for the fine level problem.

Lemma 4. ***Descent direction for the fine level function.*** *Assume that I_h^H and I_H^h are CIT operators, that F_H satisfies Definition 15, and that m gradient steps have been computed at coarse level. Then, $I_H^h(x_{H,m} - x_{H,0})$ is a descent direction for $L_h + R_h$ at x_h.*

Proof. Set $x_h \in \mathbb{R}^{N_h}$ and let us define $p_H := x_{H,m} - x_{H,0}$. Recall that $x_{H,0} = I_h^H x_h$. From the definition of descent direction we have that:

$$\langle p_H, \nabla F_H(x_{H,0}) \rangle \leq 0.$$

By the first order coherence and imposing $I_h^H = \nu^{-1} \left(I_H^h\right)^T$ we obtain

$$\langle p_H, \nabla F_H(x_{H,0}) \rangle = \langle p_H, I_h^H \nabla (L_h + R_h)(x_h) \rangle = \nu^{-1} \langle I_H^h(p_H), \nabla (L_h + R_h)(x_h) \rangle \leq 0.$$

□

Minimizing the coarse model F_H thus provides a descent direction for the fine level problem by construction. We can now define a multilevel iteration. As a picture is worth a thousand words, we provide a scheme of a multilevel iteration in Figure 3.4. Formally, a multilevel step can be described as the following "inner" Algorithm 1.

Algorithm 1 MultiLevel (ML) step for a smooth functional at any given iteration

if Coarse correction at current iteration **then**
  $x_{H,0} = I_h^H x_h$
  $v_H = I_h^H \left(\nabla L_h(y_h) + \nabla R_h(x_h)\right) - \left(\nabla L_H(x_{H,0}) + \nabla R_H(x_{H,0})\right)$
  Set $\tau_H > 0$ and $\alpha_H > 0$ according to [70]
  $x_{H,m} = \underbrace{(\mathrm{Id} - \tau_H \nabla F_H) \circ \ldots \circ (\mathrm{Id} - \tau_H \nabla F_H)}_{m \text{ gradient steps}}(x_{H,0})$
  $\overline{x}_h = x_h + \alpha_H I_H^h \left(x_{H,m} - x_{H,0}\right)$
else
  $\overline{x}_h = x_h$
end if

First, if we decide to use a coarse correction, the current iterate is projected to the coarse level using the information transfer operator I_h^H to obtain $x_{H,0}$. The coarse model is also constructed with the projection of the current gradient. Then $m + 1$ gradient steps are computed on this function to obtain a new coarse iterate $x_{H,m}$. The difference between $x_{H,m}$ and $x_{H,0}$ is prolonged to the fine level using I_H^h and added to the current iterate (with maybe an additional line search to guarantee descent).

Otherwise, i.e., if we choose not to use a coarse correction, nothing changes and the algorithm continues with fine level steps.

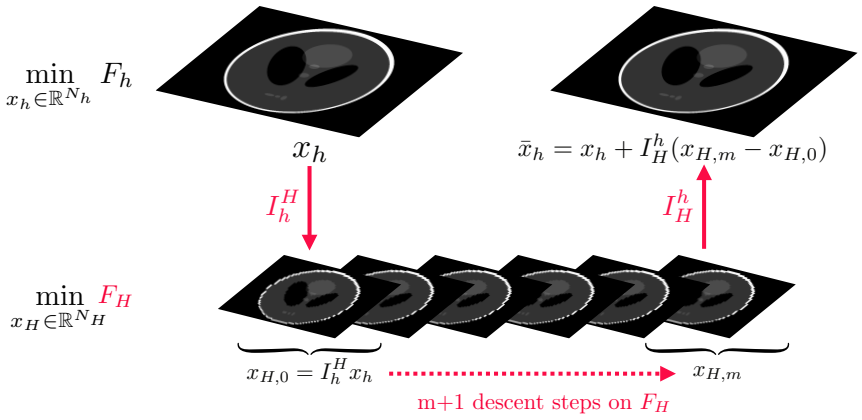

Figure 3.4: Scheme of a multilevel iteration. It encompasses the projection of the current iterate to the coarse level; the definition of a coarse level function; several minimization steps at coarse level; prolongation of the resulting coarse iterate to the fine level; and finally, the correction of the fine level iterate with the difference between the last and first coarse iterates. Highlighted in red are the elements that require careful construction for the correction of the fine level iterate to be a descent direction.

Multi-levels algorithm. The extension from a *two*-level framework to a *multi*-level framework is quite direct. The coarse model we just constructed is smooth, and a similar method can be employed to construct a coarser model (i.e., a coarse model for this coarse model) that is first order coherent with respect to the previous coarse model.

In a sense, levels work by pair. Each coarse model is used to accelerate its finer one. Thus, one can think about going from one level to the other in several manners by manipulating these pairs. For instance, the most common scheme is called the V-cycle. It consists in going all the way from the fine level to the coarsest one, and then going back to the fine level. Such V-cycle is illustrated in Figure 3.5 for 4 levels. Using the wavelet terminology, the finest level is associated with a resolution J (which corresponds to 2^J pixels), and the coarsest level is associated with a resolution $J-3$. For each pair of levels, we use similar coarse correction scheme as in Figure 3.4.

3.2.2 Literature review

We now discuss the literature on multilevel optimization methods. There exist three concurrent algorithms to the method we propose and discuss in the next chapters of this manuscript: [89], [90] and [88]. We will discuss in depth about their differences with our method in Chapter 4.

Recall that we aim to solve optimization problem formulated as the sum of two functions:

$$\hat{x} \in \operatorname*{Argmin}_{x \in \mathcal{H}} F(x) := L(x) + R(x)$$

where L is continuously differentiable.

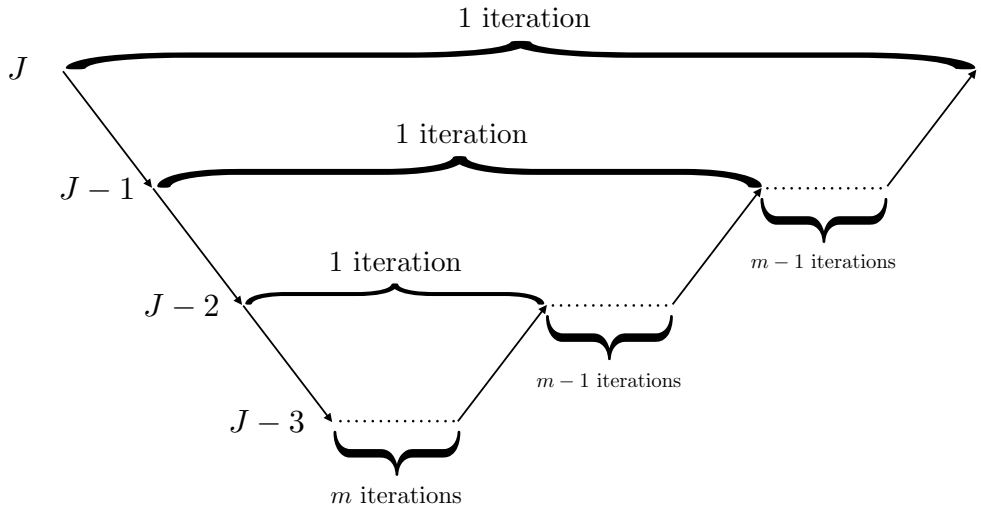

Figure 3.5: Illustration of a multilevel update with a V-cycle for a 4 levels algorithm. The algorithm passes through each level to collect its first order information before sending it to the next coarse level until reaching the coarsest one. At this point m iterations are computed at each level recursively.

Multilevel approaches have been mainly studied for the resolution of partial differential equations (PDEs), in which the underlying problem to solve can be formulated as minimizing a differentiable function [96, 98, 102]. Most of the multilevel algorithms are based on the seminal work of Nash [96] and its method MG/OPT. MG/OPT and algorithms it inspired are applied to minimize smooth objective functions by first order methods even in non-convex setting [103] and sometimes to solve PDEs with an optimization based approach [88, 104, 105].

Most contributions have been made to adapt MG/OPT to different problems and contexts. Although it may understate their contributions, experts in the field acknowledge that the greatest challenge in multilevel optimization lies in its practical implementation. In a later work [106], Nash acknowledged that practical improvements in the convergence, when using a multilevel procedure, almost entirely depend on how the algorithm is actually implemented, as the theoretical guarantees are minimal.

Let us start with extensions of the MG/OPT framework to smooth optimization for image restoration problems.

Application to (smooth) image restoration. Multilevel algorithms have been employed in many applications, such as photoacoustic tomography [107], discrete tomography [108] and phase retrieval [109]. These three applications are quite different from one another but multilevel optimization still provides acceleration in all cases.

The authors of [109] proposed a multilevel algorithm to solve an unregularized (and not necessarily convex) ptychographic phase retrieval problem, in order to work around the non-smoothness of the regularization. On large sets of experiments, they showed quite clearly the potential of multilevel algorithms.

In [107], the authors propose a multilevel algorithm to solve the inverse problem of photoacoustic tomography. At the time of publication, ℓ_1 total variation regularization was among the state-of-the-art regularizations for this problem. In order to construct

their multilevel algorithm, they smoothed the regularization term (therefore reducing the sparsity effect). They also incorporated extrapolation steps of FISTA [68] to accelerate the convergence. The authors showed that their multilevel algorithm was able to reconstruct images faster than single level algorithms in their experiments, without, however, providing a theoretical convergence analysis.

The authors of [108] propose a construction of coarse model based on Riemannian optimization techniques to propagate box constraints at fine level, to the coarse levels. The method is applied to discrete tomography problems formulated as the sum of two smooth terms with box constraints. The proposed data fidelity term is a Kullback-Leibler divergence, whose gradient is not Lipschitz, so the convergence is actually not guaranteed in this case. The definition of the coarse levels is quite involved as the first order coherence incorporates the box constraints through a Riemannian gradient. With numerical experiments, the authors demonstrate that incorporating the box constraint at coarse level greatly improves the reconstruction speed.

Smooth objective functions, and in particular smooth regularizations are not state-of-the-art in a lot of image reconstruction settings, so an extension to the non-smooth case for multilevel optimization is needed to reach state-of-the-art reconstructions.

Non-smooth multilevel optimization. There exist three papers that proposed construction of multilevel algorithm for non-smooth optimization problems.

In [89], the authors propose MAGMA: an accelerated multilevel proximal gradient algorithm for convex optimization problems. The iterations are computed using a combination of inertia, proximal projection and mirror gradient descent. A rate of convergence of $1/k^2$ is obtained for the objective function values with a slightly worse constant than for its single level counterpart.

The same construction, but for forward-backward updates in a potentially non-convex setting was later proposed in [90].

The rate of convergence is described as "optimal" for the class of functions in [89], but the numerical experiments do not really show clearly the impact of the multilevel steps. For instance, while the hyperparameters of the multilevel algorithm, such as the number of iterations at the coarse level or the number of levels, are mentioned, they are not discussed in depth. Similarly, the construction of the first-order coherence could have been elaborated. Overall, some decisions in the algorithm have an unshown impact on the convergence.

Nevertheless, these works were the first attempts to introduce multilevel methods in non-smooth optimization, and they introduced key concepts such as the smoothing of R to obtain first order coherence between levels. We will build upon these concepts in this manuscript.

More recently, authors of [88] proposed to construct adaptive restriction operators to alleviate the subgradient set-value complexity. They work as follows: if one uses a multilevel correction at points whose subgradient is set-valued, the adaptive restriction operators will reduce this set to a singleton (like a gradient) that will be used to define the first order coherence. As we will see it in Chapter 4 this method requires strong convexity assumption on L to benefit from additional convergence properties with respect to the work done by authors of [89, 90], and subsequently by us in this manuscript.

Now, we present multilevel methods that have been applied to non-smooth optimiza-

tion, but in contexts that are not directly relevant to image restoration. In [110], the authors presented two multilevel versions of the Frank-Wolfe algorithm and the Inexact Augmented Lagrangian Multiplier method for the recovery of low-rank matrices through principal component pursuit. The proposed multilevel method is constructed assuming that the low-rank components of the matrix can be recovered exactly at coarse levels (which seems to be valid for rank-one matrices). A similar approach was unfolded to create a multilevel algorithm in [111] that is able to solve semidefinite programming relaxation of polynomial optimization problem. It was applied to solve efficiently PDEs problem.

Higher order multilevel optimization method. Higher order optimization methods come with faster convergence rates and higher iteration costs. Newton-like optimization method already had been accelerated using multigrid approaches to solve linear system [97]. The ideas of Nash can naturally be extended to higher order optimization methods by increasing the order of the coherence between levels [102, 112].

Choice of information transfer operators. Beyond the obvious coherence property between the information transfer operators, one can ask: is there optimal choices? If not, is there clearly better choices than others?

In an image restoration context, it was proposed to study the way information transfer operators would preserve algebraic properties of the operator A when constructing the so-called Galerkin approximation. Notably, in [113, 114], the authors discuss algebraic ways of defining multilevel algorithms so that the coarse levels are proven to be computationally efficient to use with respect to the fine level. The idea is to look at the property of common convolution matrices (Toeplitz, circulant, etc.) when decimated and multiplied by other convolution matrices (which is a general way of defining information transfer operators.) In particular, the use of the Haar wavelet basis to define the restriction and prolongation operators preserves the Toeplitz structure of the convolution matrix [113,114]. Preserving this structure allows using fast transform to compute the matrix vector product at both fine and coarse levels [115].

Similar ideas have been used in later works such as [116] where the problems to be solved incorporates a Tikhonov based regularization. This prompts a way to define a distance from the current iterate to the true solution in the same manner as the residual for PDEs.

If we revert to the more studied context of solving linear problems ($\mathrm{A}x = b$), the choice of information transfer operators has been formulated as an optimization problem: how much should one reduce the dimension with respect to the residual of the problem? This question can be actually shown to be a NP-hard problem [117]. Nevertheless, there exist good solutions. As this literature is quite large, and to not distract the reader too much, we present only approaches we thought were worth considering in our context: learning based methods. In [118], the authors train a neural network to minimize the Frobenius norm of the prolongation matrix that send the residual from coarse to fine scales. Assuming that at coarse level the linear system is solved exactly, the authors show that such network could generalize to other linear systems. A similar idea was tested in a reinforcement learning context in [119].
You can find more references on multilevel algorithms in optimization in Appendix A.1.

3.2.3 Main obstacles to multilevel methods in optimization

In this section, we explore two key obstacles to the application and effectiveness of multilevel methods in optimization: the "Frequency Principle", and the fact that a decrease in the objective function through multilevel steps does not necessarily ensure the convergence of the iterates.

The Frequency Principle. In most optimization applications, iterative methods typically reduce low-frequency residuals before addressing high-frequency ones. This phenomenon has long been observed in image restoration across various algorithms [120] to our own experiments (see Chapter 5 and 6). Similarly, during the training of neural networks, a similar pattern is observed, known as the Frequency Principle (F-principle) [121] or the spectral bias of neural networks [122]:

Deep Neural Networks often fit target functions starting from low frequencies and gradually moving to high frequencies during training.

The studies in [121, 122] have demonstrated this behavior on standard image classification datasets like MNIST [123] and CIFAR-10 [124], where neural networks first learn the low-frequency components of the target function, followed by the high-frequency components. While certain neural network architectures may not exhibit this bias [125], these architectures often lack practical utility [125].

This behavior contrasts sharply with the numerical solving of partial differential equations (PDEs), where high frequencies are corrected first, thereby fully justifying the use of multigrid methods in that context.

The Frequency Principle suggests that multilevel algorithms in most optimization scenarios are unlikely to offer a universal accelerator as current optimization algorithms are also able to recover to find the low frequencies of the solution, emphasizing the importance of careful and robust algorithm design [106].

Decrease of the objective function with ML steps is not sufficient for convergence of the iterates. In these last paragraphs, we show, with a counter-example, that decreasing the objective function with multilevel (ML) steps does not ensure convergence of the iterates. This analysis fits the framework of MG/OPT [96]: the proof of convergence relies on intertwining multilevel steps with gradient steps, the latter ensuring the convergence of the whole sequence. Hence, this analysis calls for a different proof of convergence, or a different construction of multilevel algorithms to recover state-of-the-art guarantees (i.e., convergence to a minimizer). Consider the following setup inspired by the example discussed in [88].

Lemma 5. *Let $N \in \mathbb{N}$. Denote by $F : \mathbb{R}^N \to \mathbb{R}$ a non-convex function to optimize. Suppose we have an update map $\sigma : \mathbb{R}^N \to \mathbb{R}^N$ that generates a sequence $\{x_{k+1}\} = \sigma(x_k)$, which is assumed to converge from any starting point to a critical point of F. Additionally, suppose there exists an operator ρ such that $F(\rho(x)) \leq F(x)$ for all x. Despite these two assumptions, the sequence generated by intertwining ρ updates with σ updates may not converge to a critical point of F.*

Proof. To illustrate this, consider the non-convex function $F : \mathbb{R}^2 \to \mathbb{R}$ defined as follows:

$$F(x^1, x^2) = \begin{cases} \frac{1}{1+x_2} & \text{if } |x^1| \geq 1, \\ (0,0) \text{ is a minimizer} & \text{if } |x^1| < 1. \end{cases}$$

Define the update maps:

- $\sigma(x^1, x^2) = \frac{9}{10}(x^1, x^2)$,
- ρ such that $\rho(x^1, x^2) = \left(\frac{10}{9}x^1, x^2\right)$ for $|x_1| \geq 1$.

Both σ and ρ satisfy the assumptions:

$$F(\sigma(x^1, x^2)) \leq F(x^1, x^2) \quad \text{and} \quad F(\rho(x^1, x^2)) \leq F(x^1, x^2).$$

However, alternating between ρ and σ updates does not guarantee convergence. Instead, it can cause the sequence to stall. To see that, suppose that we start from the point $(x^1, x^2) = (1, 1)$. The sequence generated by alternating between ρ and σ updates is as follows:

$$\begin{aligned} (x^1, x^2) &= (1, 1), \\ \rho(x^1, x^2) &= \left(\frac{10}{9}, 1\right), \\ \sigma\left(\frac{10}{9}, 1\right) &= \left(\frac{9}{10} \cdot \frac{10}{9}, 1\right) = (1, 1), \\ &\vdots \end{aligned}$$

The sequence oscillates indefinitely around $(1, 1)$, which concludes the proof □

This illustrates that ensuring a decrease in the objective function alone is insufficient for the convergence of the iterates. Such result can also be obtained with convex function, as shown in the following.

Lemma 6. *Let $N \in \mathbb{N}$. Denote by $F : \mathbb{R}^N \to \mathbb{R}$ a convex function to optimize. Suppose we have an update map $\sigma : \mathbb{R}^N \to \mathbb{R}^N$ that generates a sequence $\{x_{k+1}\} = \sigma(x_k)$, which is assumed to converge from any starting point to a minimizer of F. Additionally, suppose there exists an operator ρ such that $F(\rho(x)) \leq F(x)$ for all x. Despite these two assumptions, the sequence generated by intertwining ρ updates with σ updates may not converge to a minimizer of F.*

Proof. Consider the function $F : \mathbb{R}^2 \to \mathbb{R}$ defined as:

$$F(x^1, x^2) = \begin{cases} \frac{1}{2}\|(x^1, x^2)\|^2 - \frac{1}{2} & \text{if } \|(x^1, x^2)\|^2 \geq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Now, define the update maps:

- $\sigma(x^1, x^2) = \frac{1}{2}(x^1, x^2)$ if $\|(x^1, x^2)\|^2 \geq 1$; otherwise, $\sigma(x^1, x^2) = (x^1, x^2)$.

- $\rho(x^1, x^2) = -(x^1, x^2)$.

The function F has a minimum value of 0, and $\sigma(\cdot, \cdot)$ and $\rho(\cdot, \cdot)$ satisfy the conditions for decreasing F at each step. However, the sequence generated by alternating σ and ρ may not converge.

Suppose that we start from the point $(x^1, x^2) = (2, 2)$. The iterations read

$$\begin{aligned}
(x^1, x^2) &= (2, 2), \\
\rho(x^1, x^2) &= (-2, -2), \\
\sigma(-2, -2) &= (-1, -1), \\
\rho(-1, -1) &= (1, 1), \\
\sigma(1, 1) &= (1, 1), \\
\rho(1, 1) &= (-1, -1), \\
\sigma(-1, -1) &= (-1, -1), \\
&\vdots
\end{aligned}$$

The sequence will oscillate indefinitely, despite the function values having reached the minimum value. □

This example clearly underscores that additional coherence in the algorithm is necessary to ensure convergence of the iterates.

Remark 4. *In practice, multilevel (ML) steps generally perform much better due to first-order coherence, which guarantees coherence between the updates produced by σ that would be the gradient descent and the updates produced by ρ that would be our multilevel step.*

3.3 Conclusion

In this chapter, we have presented and discussed the motivation behind applying multilevel methods to optimization. The theoretical and practical success of multigrid methods in the solving of PDEs are such that these methods are now considered state-of-the-art for most of the PDE problems they are applied to.

Even though multilevel optimization has shown promising practical result on a wide range of optimization problem; the theoretical understanding and development of these methods is not on par with the one of multigrid methods. Moreover, a lot of optimization problems are not yet amenable to multilevel optimization, in particular when considering imaging problem that involves non-smooth regularizations, whose proximity operator may not be available explicitly. In the rest of this manuscript will try to close this gap.

Part II

IML FISTA: theory and applications

Chapter 4

IML FISTA: a new framework for non-smooth multilevel optimization

In this chapter we present the central contribution of this thesis, Inexact MultiLevel FISTA (IML FISTA): a multilevel algorithm with state-of-the-art convergence guarantees for non-smooth and non-proximable[1] optimization problems.

The content of this chapter was partially published in the following papers [126–128]. The presentation of the concepts we used to define IML FISTA contains more details than in [126].

4.1 Introduction

Starting with the work of Nash [96], numerous extensions of the multigrid framework to smooth optimization have been made since [103, 106–109]. One thing that we can remark when studying these extensions is that the multilevel approach can greatly accelerate the solution of optimization problems in various contexts [88–90, 106, 129, 130], and Chapter 2, Section 3.2.2.

In the context of image restoration, several constructions of smooth multilevel algorithms have been shown to significantly accelerate the solution of restoration problems. However, the final reconstruction result was not as good as it could be: to build such multilevel algorithms, the original non-smooth function to be optimized was either smoothed [107, 108], the regularization completely removed [109], or not the best available [90]. Indeed, in imaging applications, non-smooth optimization provides state-of-the-art regularization techniques. With the development of wavelet thresholding [22] and total variation denoising [13] to denoise smooth signals with sharp edges, and later compressive sensing [21, 131], it has been common and nearly ubiquitous for the last twenty years to impose sparsity in some form on the solution. Smooth penalties are notably unable to recover sparse solution, and state-of-the-art regularization such as NLTV are not proximable explicitly [33]. Thus, one can wonder how the multilevel framework would perform in a non-smooth, non-proximable optimization context to tackle imaging problems.

[1] In the sense that the proximity operator is not explicitly known.

Non-smoothness is a challenge to define multilevel algorithms. To better capture the difficulty arising in this context, we highlight in red in the algorithm that computes a coarse correction for smooth functionals (the **ML** step) (Algorithm 2), the elements that are to be adapted in the non-smooth case. A diagram of the **ML** step is displayed in Figure 4.1. For instance, the definition of the first order coherence involves the projection

Algorithm 2 MultiLevel (**ML**) step for a smooth function. What needs to be adapted for non-smooth functions is in red.

if Coarse correction at current iteration **then**
  $x_{H,0} = I_h^H x_h$
  $v_H = I_h^H \left(\nabla L_h(y_h) + \nabla R_h(x_h)\right) - \left(\nabla L_H(x_{H,0}) + \nabla R_H(x_{H,0})\right)$
  Set $\tau_H > 0$ and $\bar{\tau}_H > 0$ according to [70]
  $x_{H,m} = \underbrace{(\mathrm{Id} - \tau_H \nabla F_H) \circ \ldots \circ (\mathrm{Id} - \tau_H \nabla F_H)}_{m \text{ gradient steps}}(x_{H,0})$
  $\overline{x}_h = x_h + \bar{\tau}_H I_H^h \left(x_{H,m} - x_{H,0}\right)$
else
  $\overline{x}_h = x_h$
end if

of gradients from the fine level to the coarse level (Chapter 1, Definition 16), something we cannot do in the non-smooth case.

This chapter is dedicated to the definition of IML FISTA, an algorithm that aims to tackle non-smooth optimization problems using a multilevel approach, with state-of-the-art convergence guarantees. A proximal multilevel algorithm able to tackle problems of the form (2.7) will iterate the following steps:

$$\bar{x}_{h,k} = \mathbf{ML}(x_{h,k}), \tag{4.1}$$
$$x_{h,k+1} = \mathrm{prox}_{\tau R}(\bar{x}_{h,k} - \tau \nabla L(\bar{x}_{h,k}). \tag{4.2}$$

IML FISTA will take the following form. We highlight in red the elements we added to the standard multilevel algorithm in order to obtain state-of-the-art convergence guarantees:

$$\bar{y}_{h,k} = \mathbf{ML}(y_{h,k}), \tag{4.3}$$
$$x_{h,k+1} \approx \mathrm{prox}_{\tau R}(\bar{y}_{h,k} - \tau \nabla L(\bar{y}_{h,k})), \tag{4.4}$$
$$y_{h,k+1} = x_{h,k+1} + \alpha_{h,k}(x_{h,k+1} - x_{h,k}) \tag{4.5}$$

where $\approx$ in Equation (4.4) indicates potential errors on the proximity operator of R. To be convergent and efficient, IML FISTA needs to handle the following challenges.

Challenge 1: extending first order coherence to non-smooth functions. There exists, to the best of my knowledge, two main ways to define the first order coherence for non-smooth fine and coarse level functions. The first one, and the one we investigated: smooth both fine and coarse level functions to compute their gradients. The second one, and most recent, consists in extending the notion of coherence to subgradients [88]. However, it suffers from a lack of generality and other hindrances. We will discuss this at the end of the chapter when comparing our frameworks to others.

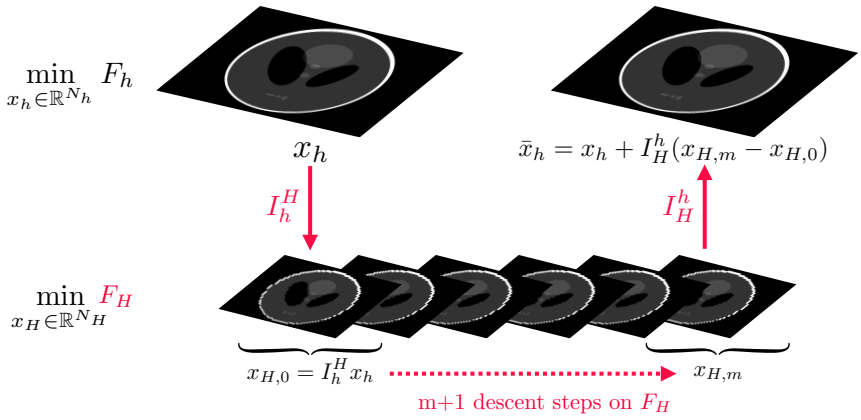

Figure 4.1: Scheme of a multilevel iteration. In this chapter, we will discuss how to deal with non-smooth and non-proximable objective functions F_h, in a multilevel setting. The elements in red require adaptation for non-smooth optimization. We will defer to Chapter 5 and Chapter 6 the discussion on the information transfer operators. The main building blocks of our algorithm are the smoothing tools (to define F_H), the inexact proximity operator, and the extrapolation steps (to reach optimal convergence rates).

Challenge 2: dealing with non-proximable penalties. State-of-the-art regularizations are not only non-smooth, but often also non-proximable. Such penalties require the computation of an estimation of the proximity operator at each iteration, which raises new questions to define an efficient multilevel algorithm. In particular, this will have an impact on the smoothing strategy.

Challenge 3: extrapolation steps. In order to reach the optimal convergence of rate of $O(1/k^2)$, we need to add extrapolation steps to our algorithm. Moreover, the **ML** step should be applied to the extrapolated point $y_{h,k}$. These steps are taken from the FISTA algorithm [68–70]. The extrapolation steps will be defined as in [70], so that along with the optimal convergence rate, we also keep the convergence to a minimizer of the objective function.

Organization of the chapter. This chapter is organized as follows. We will describe the building blocks of our algorithm in details. First, we begin by a presentation of the smoothing tools that are used to define our multilevel algorithm. Then, we will discuss how to deal with non-proximable penalties, and the influence of the error on the convergence guarantees of the algorithm. The last building block of our algorithm is the possibility of adding extrapolation steps after multilevel and proximal-gradient steps. This framework is the one of inertial algorithms such as FISTA [68–70]. Combined with multilevel steps, these extrapolation steps will help us obtain optimal theoretical convergence guarantees and practical acceleration of the minimization. We then introduce our algorithm IML FISTA in full and its proof of convergence. We will conclude this chapter with a comparison of this general framework with concurrent ones of the literature.

4.2 Smoothing to bridge the gap

We have seen in the previous chapter how to define multilevel algorithms for smooth optimization. The construction of these algorithms can be extended to non-smooth optimization problems by smoothing the fine and coarse functions. *We want to emphasize that the problem we solve is still the non-smooth one, and the smoothing is only a proxy to define the multilevel steps.*

4.2.1 Smoothing tools

The rich understanding of smooth optimization by the community has lead to the development of smoothing frameworks that can be applied to solve non-smooth optimization problems.

There exists several ways of defining a smoothed version of a non-smooth function. The one we used is the one developed by the authors of [132]. A complete presentation requires the following definitions.

Definition 17. ***Infimal convolution [53, Chapter 12].*** *Let g and ω be functions both from $\mathcal{H}$ to $(-\infty, +\infty]$. The infimal convolution of g and ω is*

$$g \square \omega : \mathcal{H} \to (-\infty, +\infty] : x \mapsto \inf_{y \in \mathcal{H}} (g(y) + \omega(x - y)) \tag{4.6}$$

This infimal convolution enjoys the following properties if g, and ω belong to $\Gamma_0(\mathcal{H})$:

Proposition 1. *Let g and ω be functions belonging to $\Gamma_0(\mathcal{H})$. Then the following properties hold:*

1. *$g \square \omega$ is convex [53, Proposition 12.11].*
2. *If g is coercive and ω is bounded below then $g \square \omega$ is everywhere unique and $g \square \omega \in \Gamma_0(\mathcal{H})$ [53, Proposition 12.14].*

The most notable example of the infimal convolution is the Moreau envelope.

The Moreau envelope. The Moreau envelope of a function g was first introduced by Moreau in his seminal works on duality and proximal point operators [133, 134]. It is a particular instance of the infimal convolution between functions.

Definition 18. ***Moreau envelope. [53, Chapter 12].*** *Let $g : \mathcal{H} \to (-\infty, +\infty]$ and let $\gamma > 0$. The Moreau envelope of g of parameter γ is defined as:*

$$^{\gamma}g = g \square \left(\frac{1}{2\gamma} \| \cdot \|^2 \right). \tag{4.7}$$

Again, if g belongs to $\Gamma_0(\mathcal{H})$, its Moreau envelope enjoys useful properties involving the proximity operator (Definition 13):

Proposition 2. ***[53, Remark 12.24].*** *Let $g \in \Gamma_0(\mathcal{H})$, and $\gamma > 0$. Then for $x \in \mathcal{H}$:*

$$ {}^{\gamma}g(x) = g(\text{prox}_{\gamma g}(x)) + \frac{1}{2\gamma}\|x - \text{prox}_{\gamma g}(x)\|^2. \tag{4.8} $$

$\text{prox}_{\gamma g}(x)$ has been defined in Chapter 2, Definition 13 and is the unique point satisfying Equation (4.7), i.e., the minimum of the associated infimal convolution problem.

Proposition 3. ***[53, Proposition 12.30].*** *Let $g \in \Gamma_0(\mathcal{H})$, and $\gamma > 0$. Then ${}^{\gamma}g : \mathcal{H} \mapsto \mathbb{R}$ is differentiable on $\mathcal{H}$ with $1/\gamma$-Lipschitz continuous gradient:*

$$ (\forall x \in \mathcal{H}) \quad \nabla {}^{\gamma}g(x) = \frac{1}{\gamma}\left(x - \text{prox}_{\gamma g}(x)\right) \tag{4.9} $$

Thus, a non-smooth function g with proximity operator known under closed form possess a smooth counterpart whose gradient is easily computable with the proximity operator. Now that we have introduced the infimal convolution and the Moreau envelope, we will detail the properties of the resulting smooth approximation of g.

4.2.2 Smoothable convex function

To properly characterize the quality of the smoothing whether it is from the Moreau envelope or from another technique, the authors of [132] introduced the concept of smoothable convex function, which provides a lower bound and an upper bound on the smooth approximation. These two bounds allow us to control the tightness of the approximation.

Definition 19. ***Smoothable convex function [132].*** *Let $g \in \Gamma_0(\mathcal{H})$. Let $X \subset \mathcal{H}$ be a closed convex set. The function g is called $(\mu, \eta, K)-$smoothable over X if there exist η_1, η_2 satisfying $\eta_1 + \eta_2 = \eta > 0$ such that for every $\gamma > 0$ there exists a continuously differentiable convex function $g_\gamma : \mathcal{H} \to (-\infty, +\infty]$ such that the following hold:*

1. *for every $x \in X$, $g(x) - \eta_1\gamma \leq g_\gamma(x) \leq g(x) + \eta_2\gamma$.*
2. *The function g_γ has a Lipschitz gradient over X with Lipschitz constant which is less than or equal to $K + \frac{\mu}{\gamma}$.*

The function g_γ is called a "$\gamma-$smooth approximation" of g over X with parameters (μ, η, K).

Remark 5. *A smoothable convex function satisfies the following properties:*

1. *The sum of two smoothable convex functions on X is a smoothable convex function on X, where the parameters μ, η, K are the sum of the parameters of the two functions [132, Lemma 2.1].*
2. *The composition of a smoothable convex function on X with a linear transformation (i.e. $x \mapsto Ax + b$, with A a linear operator and b a vector in the image of A) is*

a smoothable convex function on a transformation over $A^{-1}(X-b)$ *[132, Lemma 2.2]. Where* A^{-1} *is the inverse linear mapping of* A.

Remark 6. *The Moreau envelope of* g *of parameter* $\gamma > 0$ *provides a* $(\mu, \eta, K)-$*smooth approximation of* g *over* X *with* $\mu = 1$*, and* $K = 0$*.* η *depends on the subgradients of* g *as we will see below.*

This framework was developed to guarantee that minimizing a $\gamma-$smooth approximation of a non-smooth function g, for a well-chosen γ [132, Theorem 3.1], yields an ϵ-approximation of the minimum value of g over X in a finite number of iterations with fast smooth optimization algorithms. Theorem 3.1 of [132] explicitly links the parameter γ and the number of iterations necessary to reach an $\epsilon-$approximation of the minimum value. For more details about these results and its potential use in our context, see Appendix A.2.1.

To define a smooth approximation, the authors of [132] extend the ideas behind the Moreau envelope and consider the infimal convolution of g with a continuously differentiable convex function ω.

Definition 20. ***Inf-conv*** γ***-smooth approximation [132, Definition 4.2].*** *Let* $g : \mathcal{H} \to (-\infty, +\infty]$ *be a closed proper convex function and let* $\omega : \mathcal{H} \to \mathbb{R}$ *be a* $C^{1,1}$ *convex function with Lipschitz gradient of constant* $1/\sigma$ *(*$\sigma > 0$*). Suppose that for any* $\gamma > 0$ *and any* $x \in \mathcal{H}$*, the following infimal convolution is finite:*

$$g_\gamma^{ic}(x) = \inf_{u \in \mathcal{H}} \left\{ g(u) + \gamma\omega\left(\frac{x-u}{\gamma}\right) \right\} \tag{4.10}$$

Then g_γ^{ic} *is called the inf-conv* $\gamma-$*smooth approximation of* g.

From this definition, and using the dual formulation of Equation (4.10), one can derive the following result:

Lemma 7. ***[132, Lemma 4.2].*** *Consider the setting of Definition 20 and let* X *be a closed convex set of* $\mathcal{H}$*. Suppose that* g *is subdifferentiable over* X*. Then for any* $\gamma > 0$ *and* $x \in X$ *the following holds:*

$$g(x) - \gamma\omega^*(\boldsymbol{d}_x) \leq g_\gamma^{ic}(x) \leq g(x) + \gamma\omega(\mathbf{0}) \tag{4.11}$$

where $\boldsymbol{d}_x \in \partial g(x)$*, and* ω^* *is the Fenchel conjugate of* ω.

In Equation (4.11), one can recognize the constant η_1 and η_2 in the definition of a smoothable convex function. In particular, η_1 can be chosen so that

$$\eta_1 = \sup_{x \in X} \sup_{\mathbf{d} \in \partial g(x)} \omega^*(\mathbf{d}_x) < +\infty, \tag{4.12}$$

while $\eta_2 = \omega(\mathbf{0})$. The choice of ω will thus control the tightness of the approximation. Ideally, one wants to find $\widehat{\omega}$ such that:

$$\widehat{\omega} \in \operatorname*{arg\,min}_{\omega \in C^{1,1}(\mathcal{H}, \mathbb{R})} \left\{ \sup_{x \in X} \sup_{\mathbf{d} \in \partial g(x)} \omega^*(\mathbf{d}_x) \right\} \tag{4.13}$$

This problem is rather complicated to solve, but for some known smoothing techniques such as the Moreau envelope, and for particular functions g, η is available explicitly.

For instance take g as the ℓ_1-norm over $\mathbb{R}^N$. Its Moreau envelope ${}^{\gamma}g$ yields a $(\mu = 1, \eta = N/2, K = 0)-$smooth approximation of g [132, Example 4.2]. Choosing instead the approximation $g_\gamma : x \mapsto \sum_{i=1}^{N} \sqrt{\gamma^2 + x_i^2}$, yields a $(1, N, 0)-$smooth approximation of g [132, Example 4.6]. This smoothing is significantly worse, in theory, than the Moreau envelope smoothing.

Remark 7. *Any convex function is smoothable over closed convex sets provided that its subgradients are bounded [132, Corollary 4.1].*

As presented in Algorithm 2, the first order coherence relied on the gradient of L and R, at fine and coarse levels. When considering non-smooth functions, such a definition is not possible anymore. This smoothing framework offers us a new way to handle the first order coherence and ensure that decreasing the coarse level will decrease the fine level objective function.

4.3 Inexact proximity operator: estimation and guarantees

As previously discussed in Chapter 2, Section 2.1.2, state-of-the-art regularizations considered in variational approaches are often the result of the composition of a linear operator D and a non-smooth penalty g. The proximity operator of the composition is explicitly available first if the proximity operator of g is known under closed form, and second if D and its adjoint respect the following relationship: $\mathrm{D}^*\mathrm{D} \propto \mathrm{Id}$. This encompasses orthogonal operators such as wavelet transform but appears limited as many standard choices of D do not satisfy this property. If D encodes the finite difference operator involved in the total variation, then this relationship is not verified anymore, and the proximity operator of $g \circ \mathrm{D}$ needs to be estimated through an optimization procedure. This estimation has been investigated rigorously in the literature [27, 62, 70, 135–138].

To account for inexactness in the proximity operator computation and adapt convergence proofs, one needs to enlarge the notion of subdifferential through the following definition [62, 70, 135]:

Definition 21. ***ϵ-subdifferential.*** *The ϵ-subdifferential of R at $z \in$ dom R is defined as:*

$$\partial_\epsilon R(z) = \{y \in \mathbb{R}^N \mid R(x) \geq R(z) + \langle x - z, y\rangle - \epsilon, \forall x \in \mathbb{R}^N\}. \tag{4.14}$$

We used three types of approximations of proximity operators that were proposed in the literature [27, 62, 135], based on this definition.

Definition 22. ***Type 0 approximation [27].*** *We say that $z \in \mathbb{R}^N$ is a type 0 approximation of $\mathrm{prox}_{\gamma R}(y)$ with precision ϵ, and we write $z \approx_{0,\epsilon} \mathrm{prox}_{\gamma R}(y)$, if and only if:*

$$\|z - \mathrm{prox}_{\gamma R}(y)\| \leq \sqrt{2\gamma\epsilon}. \tag{4.15}$$

This approximation is the easiest to formulate with respect to the forward-backward step. The sequence is written as follows:

$$x_{k+1} = \text{prox}_{\gamma R}(x_k - \gamma \nabla L(x_k)) + e_k, \tag{4.16}$$

where e_k models the error in the computation of the proximity operator, and is such that $\|e_k\|^2 = \epsilon$. To obtain convergence of the sequence, the summability of the norm of the error is sufficient [27, Theorem 3.4].

The next two types of approximations were proposed in [62] to estimate up to a precision ϵ the proximity operator.

Definition 23. ***Type*** 1 ***approximation [62].*** *We say that $z \in \mathbb{R}^N$ is a type 1 approximation of $\text{prox}_{\gamma R}(y)$ with precision ϵ, and we write $z \approx_{1,\epsilon} \text{prox}_{\gamma R}(y)$, if and only if:*

$$0 \in \partial_\epsilon \left(R(z) + \frac{1}{2\gamma}\|z - y\|^2 \right). \tag{4.17}$$

Definition 24. ***Type*** 2 ***approximation [62].*** *We say that $z \in \mathbb{R}^N$ is a type 2 approximation of $\text{prox}_{\gamma R}(y)$ with precision ϵ, and we write $z \approx_{2,\epsilon} \text{prox}_{\gamma R}(y)$, if and only if:*

$$\gamma^{-1}(y - z) \in \partial_\epsilon R(z). \tag{4.18}$$

Remark 8. *Approximations of type 2 imply approximations of type 1 [62, 70] and under some conditions discussed in [62], approximation of type 0 implies approximation of type 2. Note that these three types of approximations are not equivalent.*

When these approximations are used in forward-backward-based algorithms, convergence of the sequence to a minimizer is known from the literature: approximations of type 1 and 2 are covered by [70] for inertial versions of the forward-backward algorithm, while the type 0 approximation is treated in [27] only for the forward-backward algorithm. Typical cases of image restoration, where dual optimization is used, are based on approximations of type 2 (see Chapter 5). We will use only approximations of type 2 in our experiments, following [62], but the theoretical results are valid and presented for the three types of approximations.

The type of chosen approximation defines how the sequence $(\epsilon_k)_{k\in\mathbb{N}}$ will be summable against k or k^2. Therefore, it is independent, theoretically, of the multilevel framework.

4.3.1 Computation of the proximity operator of $g \circ \mathrm{D}$

If $\mathrm{D} : \mathbb{R}^N \mapsto \mathbb{R}^K$ is not a projection on a tight frame (e.g., a union of wavelets) or an orthogonal basis, a common way of estimating the proximity operator consists in formulating the minimization problem in the *dual*. Denoting $R = g \circ \mathrm{D}$, we have that (see for instance [62, 63, 139, 140]):

$$(\forall x \in \mathbb{R}^N) \qquad \text{prox}_{\gamma R}(x) := \text{prox}_{\gamma g \circ \mathrm{D}}(x) = x - \mathrm{D}^* \widehat{u}, \tag{4.19}$$

with:

$$\widehat{u} \in \underset{u \in \mathbb{R}^K}{\text{Argmin}} \frac{1}{2}\|\mathrm{D}^* u - x\|^2 + (\gamma g)^*(u), \tag{4.20}$$

where g^* is the convex conjugate of g. This problem is known as the dual problem. One can directly see that the linear operator is now inside the smooth term, and the composition has an explicit gradient formulation.

An approximation of $\widehat{u}$ may be obtained by applying any adapted optimization method to (4.20). For instance, FISTA yields the following sequence (choosing $u_0 = v_0$):

$$u_{k+1} = \left(\mathrm{Id} - \gamma \mathrm{prox}_{g/\gamma}\left(\cdot/\gamma\right)\right)\left((\mathrm{Id} - \mathrm{DD}^*)v_k + \gamma \mathrm{D}x\right) \tag{4.21}$$

$$v_{k+1} = (1+\alpha_k)u_{k+1} - \alpha_k u_k. \tag{4.22}$$

where the first step is deduced from the Moreau decomposition [53] (that links the proximity operator of g to the proximity operator of g^*). Dual optimization is a simple way to estimate the proximity operator while offering guarantees on the computed approximation, as stated in the following lemma.

Proposition 4. ***Dual optimization yields approximation of type 2.*** *Assume that $(u_k)_{k\in\mathbb{N}}$ is a minimizing sequence for the dual function in (4.20). This yields:*

- *A convergent sequence $(x - \mathrm{D}^* u_k)_{k\in\mathbb{N}}$ to $\mathrm{prox}_{\gamma g\circ \mathrm{D}}$ (4.19).*
- *This sequence provides a type 2 approximation of the proximity operator.*

Proof. The first point comes from [62, Theorem 5.1]. Then the approximation of type 2 comes from [62, Proposition 2.2, and 2.3]. □

At each iteration of FISTA or IML FISTA (Equation (4.4)), an estimation of the proximity operator is computed by solving approximately Problem (4.20).

4.3.2 Accuracy of the computation of the proximity operator

Convergence guarantees of algorithms using inexact proximity operators are directly linked to the decrease of the error introduced by estimating the proximity operator at each iteration. This problem was notably addressed in [141], where the authors introduced the Speedy Inexact Proximal-Gradient Strategy (SIP). In order to achieve this decrease, the number of sub-iterations used to estimate the proximity operator is dynamically increased.

More precisely, if at step k, $F(x_k) > F(x_{k-1})$, we decrease the tolerance[2] (tol) on the estimation of the proximity operator at the next steps $k+1, k+2, \ldots$ as tol controls the relative distance between two consecutive sub-iterates of the proximity operator estimation. We expect that a small value of tol will induce a high accuracy (i.e. reduce the value of ϵ) on the estimation.

This minimization is carried out by FISTA coupled with a warm start strategy as in [57]: the estimate of the proximity operator at step k is used as the initial point for the estimation at step $k+1$.

In all cases, a lower error is correlated with a higher computational cost, which is why some strategies rather use a fixed budget of sub-iterations to compute the proximity

[2]Intuitively, the error on the computation of the proximity operator acts like the best precision one can reach on the minimal value of F. It leads to divergence on the sequence when this error is reached.

Algorithm 3 Accuracy of the proximity operator estimation

```
Set x_0 ∈ ℝ^N,
for k = 0, 1, ..., do
    if F(x_k) > F(x_{k-1}) then
        tol = tol/10
    end if
end for
```

operator using the dual formulation [57]. This fixed budget comes at the cost of a limited precision on the estimated solution and may lead to divergence after many iterations.

4.3.3 Circumventing the inexactness

This presentation would not be complete without mentioning the numerous primal-dual algorithms that have been proposed to solve problems of the form (2.7).

Primal-dual algorithms. Many methods circumvent this dual optimization by directly introducing dual steps paired with primal steps to reach a minimizer [140, 142–144], but their cost for large-scale problems remains high, and they may still need to compute inexact proximity operators [145].

Inexact proximal algorithms with fixed number inner iterations. Alongside the literature we based our algorithm on, other works have focused on deriving inexact proximal algorithm where the inner loop had a fixed number of iterations (in opposition to our strategy here).

The idea is to formulate the problem in a primal-dual sense [146,147], which allows one to derive inexact forward-backward algorithms with a fixed number of inner primal-dual iterations to evaluate the proximity operator, combined with the warm start strategy.

Even though inertial versions of these algorithms do exist [147], convergence guarantees are weaker than those of FISTA with inexact proximity operator [70].

4.4 Extrapolation steps

There exist plenty of acceleration techniques in optimization. For problems of the form (2.7), the optimal worst case rate of convergence is $1/k^2$. To reach this rate of convergence, it was first proposed in [67] to add extrapolation steps to a classical gradient descent algorithm and later to proximal gradient descent in [68].

4.4.1 Our choice of extrapolations steps

We want to be able to handle inexactness in the proximity operator, and thus small perturbation errors on the sequence. Luckily, a framework developed in [70] precisely describes how to deal with errors in the computation of the proximity operator while

allowing extrapolation steps. The algorithm is as follows:

$$x_{k+1} = \mathrm{FB}_i^{\epsilon_k}(y_k), \tag{4.23}$$

$$t_{k+1} = \left(\frac{(k+1)+a-1}{a}\right)^d, \tag{4.24}$$

$$y_{k+1} = x_{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)(x_{k+1} - x_k). \tag{4.25}$$

where

$$(\forall x \in \mathbb{R}^N) \qquad \mathrm{FB}_i^{\epsilon}(x) \approx_{i,\epsilon} \mathrm{prox}_{\tau R}\left(x - \tau \nabla L(x)\right) \tag{4.26}$$

replaces the exact computation of the proximity operator. The sequence $(t_k)_{k\in\mathbb{N}}$ is parametrized by [70, Definition 3.1]:

$$\begin{cases} d = 0 \\ \text{or} \quad d \in]0,1] \ \& \ a > \max\{1, (2d)^{\frac{1}{d}}\}. \end{cases} \tag{4.27}$$

This allows us to go continuously from a forward-backward algorithm (with $d = 0$) to FISTA ($d = 1$). This parameter was primarily introduced as a way to control the inertia with respect to the error committed when estimating the proximity operator [70].

4.4.2 Inertia and approximation error

The necessary speed of the error's decrease depends on the choice of d (Equation (4.27)), therefore on the quantity of inertia incorporated at each iteration; and on the type of approximation we are using. Indeed, going from $d = 1$ (FISTA) to $d = 0$ (FB) relaxes the decrease speed [70]. This can be useful if the approximation error is too large.

We will see in our numerical experiments (Chapter 5, Section 5.6) that it is also advantageous, in some contexts, for multilevel algorithms to use $d < 1$ to avoid unnecessary oscillations when leaving the coarse models.

4.5 Inexact MultiLevel FISTA

To facilitate the presentation of the proposed algorithm IML FISTA, we will first introduce its two-levels version, then in Section 4.5.5 we will generalize it to an arbitrary number of levels.

4.5.1 Our algorithm

Following the notations introduced in Chapter 2, we index by h (resp. H) all quantities defined at the fine (resp. coarse) level. We thus define $F_h := F : \mathbb{R}^{N_h} \to (-\infty, +\infty]$ the objective function at the fine level where $N_h = N$, such that $F_h = L_h + R_h$ (with $L_h := L$ and $R_h := R$). We pair this objective function at fine level with its coarse level approximation, which is denoted $F_H : \mathbb{R}^{N_H} \to (-\infty, +\infty]$, with $N_H < N_h$, and where L_H, R_H are lower dimensional approximations of L and R.

One standard step of our algorithm can be summarized by the following three instructions:

$$\bar{y}_{h,k} = \mathrm{ML}(y_{h,k}), \tag{4.28}$$
$$x_{h,k+1} = \mathrm{FB}_i^{\epsilon_{h,k}}(\bar{y}_{h,k}), \tag{4.29}$$
$$y_{h,k+1} = x_{h,k+1} + \alpha_{h,k}(x_{h,k+1} - x_{h,k}) \tag{4.30}$$

which are developed in details in Algorithm 4, and where **ML** encompasses Steps 3 to 11.

The algorithm works as follows. Given the current iterate $y_{h,k}$ at fine level, we can decide to update it either by a standard fine step, combining Steps 10 and 12-14 of the algorithm, or by performing iterations at the coarse level (cf. steps 5-8), followed by a standard fine step (cf. steps 12-14).

Particular attention is paid to steps 5-8, which produce a coarse correction that is used to define an intermediate fine iterate $\bar{y}_{h,k}$. The coarse correction is used to update the auxiliary variable $y_{h,k}$ and not $x_{h,k}$ directly (see Equations (4.29) and (4.30)). This makes sense for two reasons: first, the iterate that receives the forward-backward update is $y_{h,k}$, therefore it is natural to update it with an ML step beforehand; second, the convergence framework we chose is suited to deal with corrections to the sequence if incorporated inside the forward-backward step.

To obtain this coarse correction, the current iterate $y_{h,k}$ is projected to the coarse level thanks to a projection operator I_h^H, and it is used as the initialization for the minimization of the coarse approximation F_H.

This generates a sequence $(x_{H,k,\ell})_{\ell\in\mathbb{N}}$, where k represents the current iteration at the fine level and ℓ indexes the iterations at the coarse level. This sequence is defined by $x_{H,k,\ell+1} = \Phi_{H,\ell}(x_{H,k,\ell})$, with $\Phi_{H,\ell}$ any operator such that, after $m > 0$ coarser iterations, $F_H(x_{H,k,m}) \leq F_H(x_{H,k,0})$. A discussion about an adequate choice for m is deferred to Chapter 5. While this operator has to implicitly adapt to the current step k, its general construction does not depend on k. After m iterations at the coarse level we obtain a coarse direction $x_{H,k,m} - x_{H,k,0}$, prolonged at the fine level with I_H^h to update $y_{h,k}$.

The central point of multilevel approaches is to ensure that the correction term $x_{H,k,m} - x_{H,k,0}$, after prolongation from the coarse to the fine level, leads to a decrease of F_h. For this, particular care must be taken in the selection of the following elements:

(i) the coarse model F_H,

(ii) the minimization scheme $\Phi_{H,\bullet}$,

(iii) the information transfer operators I_h^H and I_H^h.

We detail these choices in the following subsections.

4.5.2 Smooth coarse model for non-smooth multilevel optimization

In our algorithm the construction of coarse functions relies on smoothing the non-differentiable R_h [132] to ensure similarity with the fine model, and at the same time to impose desirable properties to the coarse model. As demonstrated in [89,90], smoothing is a natural choice to extend ideas coming from the classical smooth case [98] to multilevel proximal gradient

Algorithm 4 IML FISTA

Set $x_{h,0}, y_{h,0} \in \mathbb{R}^N$, $t_{h,0} = 1$
while Stopping criterion is not met **do**
  if Descent condition and $r < p$ Under this condition, we use coarse models **then**
    $r = r + 1$,
    $x_{H,k,0} = I_h^H y_{h,k}$ Projection
    $x_{H,k,m} = \Phi_{H,m-1} \circ .. \circ \Phi_{H,0}(x_{H,k,0})$ min F_H
    Set $\bar{\tau}_{h,k} > 0$,
    $\bar{y}_{h,k} = y_{h,k} + \bar{\tau}_{h,k} I_H^h \left(x_{H,k,m} - x_{H,k,0}\right)$ Coarse step update whose size is set by $\bar{\tau}_{h,k}$
  else
    $\bar{y}_{h,k} = y_{h,k}$
  end if
  $x_{h,k+1} = \mathrm{FB}_i^{\epsilon_{h,k}}(\bar{y}_{h,k})$ min F_h
  $t_{h,k+1} = \left(\frac{k+a}{a}\right)^d$, $\alpha_{h,k} = \frac{t_{h,k}-1}{t_{h,k+1}}$
  $y_{h,k+1} = x_{h,k+1} + \alpha_{h,k}(x_{h,k+1} - x_{h,k})$. Inertial step
end while

methods. We take the ideas originally proposed in [89, 90], and develop them further in the present contribution.

Smoothed convex approximations exist if the smoothing is done according to the principles developed in [132], and presented in Section 4.2.2, where the sum $\eta_1 + \eta_2$ depends on R and on the type of smoothing (Definition 19).

Definition 25. ***Coarse model*** F_H ***for non-smooth functions.*** *The coarse model* F_H *is defined for the point* $y_h \in \mathbb{R}^{N_h}$ *as:*

$$F_H = L_H + R_{H,\gamma_H} + \langle v_H, \cdot \rangle, \tag{4.31}$$

where

$$v_H = I_h^H \left(\nabla L_h(y_h) + \nabla R_{h,\gamma_h}(y_h)\right) - \left(\nabla L_H(I_h^H y_h) + \nabla R_{H,\gamma_H}(I_h^H y_h)\right). \tag{4.32}$$

R_{h,γ_h} *and* R_{H,γ_H} *are smoothed versions of* R_h *and* R_H *respectively, and they verify Definition 19 with smoothing parameters* $\gamma_h > 0$ *and* $\gamma_H > 0$.

Adding the linear term $\langle v_H, \cdot \rangle$ to $L_H + R_{H,\gamma_H}$ allows to impose the so-called *first order coherence* recalled in Definition 26 below.

Remark 9. *Note that if* R_h *and* R_H *are smooth by design, one can simply replace* R_{H,γ_H} *and* R_{h,γ_h} *by* R_H *and* R_h*, respectively. The construction stays otherwise the same.*

Definition 26. ***First order coherence.*** *The first order coherence between the smoothed version of the objective function* F_h *at the fine level and the coarse level objective function* F_H *is verified in a neighborhood of* y_h *if the following equality holds:*

$$\nabla F_H(I_h^H y_h) = I_h^H \nabla \left(L_h + R_{h,\gamma_h}\right)(y_h). \tag{4.33}$$

The following lemma shows that our choice of coarse model respects the first order coherence between two smoothed fine and coarse level functions.

Lemma 8. *If F_H is given by Definition 25, it necessarily verifies the first order coherence (Definition 26).*

Proof. Consider the gradient of the coarse model F_H and combine it with the definition of v_H in Equation (4.32). It yields

$$\begin{aligned} \nabla F_H(I_h^H y_h) &= \nabla L_H(I_h^H y_h) + \nabla R_{H,\gamma_H}(I_h^H y_h) + v_H, \\ &= I_h^H \left(\nabla L_h(y_h) + \nabla R_{h,\gamma_h}(y_h)\right). \end{aligned} \tag{4.34}$$

□

This condition ensures that, in the neighborhood of the current iterates $y_h = y_{h,k}$ and $I_h^H y_{h,k} = x_{H,k,0}$, smoothed versions of the fine and of the coarse level objective functions are coherent up to order one [90].

Choice of coarse iterations. The operators $\Phi_{H,\bullet}$ aim to build a sequence producing a sufficient decrease of F_H after m iterations.

Assumption 1. ***Coarse model decrease.*** *Let $(\Phi_{H,\ell})_{\ell\in\mathbb{N}}$ be a sequence of operators such that there exists an integer $m > 0$ that guarantees that if $x_{H,m} = \Phi_{H,m-1} \circ \ldots \circ \Phi_{H,0}(x_{H,0})$ then $F_H(x_{H,m}) \leq F_H(x_{H,0})$. Moreover, $x_{H,m} - x_{H,0}$ is bounded.*

Some typical choices for $\Phi_{H,\ell}$ are the gradient descent step, inertial gradient descent step, forward-backward step or inertial forward-backward step (see Chapter 5, Section 5.4 for a comparison of these operators in a multilevel context - the choice depends mostly on the intensity of degradation for image reconstruction problems). These operators guarantee that $x_{H,m} - x_{H,0}$ is a bounded (through convergence of the sequence [70]) descent direction for F_H.

Construction of information transfer operators. Going from one level to the other requires several information transfers. For this purpose recall that information transfer operators I_h^H and I_H^h are called coherent information transfer operators if there exists $\nu > 0$ such that $I_H^h = \nu(I_h^H)^T$ (Chapter 2, Definition 14).

Fine model minimization with multilevel steps. With the previous definitions of F_H, $\Phi_{H,\bullet}$ and I_h^H, the following lemmas prove that minimization at the coarse level also induces a descent direction at the fine level.

Lemma 9. ***Descent direction for the fine level smoothed function.*** *Let us assume that I_h^H and I_H^h are CIT operators and that F_H satisfies Definition 25 and that $\Phi_{H,\bullet}$ verifies Assumption 1. Then, $I_H^h(x_{H,m} - x_{H,0})$ is a descent direction for $L_h + R_{h,\gamma_h}$.*

Proof. Set $y_h \in \mathbb{R}^{N_h}$ and let us define $p_H := x_{H,m} - x_{H,0}$. Recall that $x_{H,0} = I_h^H y_h$. From the definition of descent direction we have that:

$$\langle p_H, \nabla F_H(x_{H,0})\rangle \leq 0.$$

By the first order coherence and imposing $I_h^H = \nu^{-1}\left(I_H^h\right)^T$ we obtain

$$\langle p_H, \nabla F_H(x_{H,0})\rangle = \langle p_H, I_h^H \nabla(L_h + R_{h,\gamma_h})(y_h)\rangle = \nu^{-1}\langle I_H^h(p_H), \nabla(L_h + R_{h,\gamma_h})(y_h)\rangle \leq 0.$$

□

We can now go a step further and derive a bound on the decrease of the *non-smooth* objective function at the fine level $F_h := L_h + R_h$. Following [89, 90], we search a proper step size $\bar{\tau}_h$ that avoids "too" big corrections from the coarse level by guaranteeing that:

$$(L_h + R_{h,\gamma_h})(y_h + \bar{\tau}_h I_H^h(x_{H,m} - x_{H,0})) \leq (L_h + R_{h,\gamma_h})(y_h). \tag{4.35}$$

Lemma 10. ***Fine level decrease.*** *If the assumptions of Lemma 9 hold, the iterations of Algorithm 4 ensure:*

$$F_h(y_h + \bar{\tau} I_H^h(x_{H,m} - x_{H,0})) \leq F_h(y_h) + (\eta_1 + \eta_2)\gamma_h. \tag{4.36}$$

Proof. This directly comes from the definition of a smoothed convex function (Definition 19). As there exists a value of $\bar{\tau}_h$ satisfying Equation (4.35), we have:

$$\begin{aligned} F_h(y_h + \bar{\tau}_h I_H^h(x_{H,m} - x_{H,0})) &\leq (L_h + R_{h,\gamma_h})(y_h + \bar{\tau}_h I_H^h(x_{H,m} - x_{H,0})) + \eta_1\gamma_h \\ &\leq (L_h + R_{h,\gamma_h})(y_h) + \eta_1\gamma_h \\ &\leq F_h(y_h) + (\eta_1 + \eta_2)\gamma_h. \end{aligned} \tag{4.37}$$

□

This result shows that a coarse level minimization step leads to a decrease of F_h, up to a constant $(\eta_1 + \eta_2)\gamma_h$ that can be made arbitrarily small by driving γ_h to zero.

This type of result is commonly found in the literature of multilevel algorithms [89, 90, 127, 128], but it is not sufficient to guarantee the convergence of the generated sequence. In the next section we derive stronger convergence guarantees.

4.5.3 Non-smooth coarse model for non-smooth multilevel optimization

The previous section showed the construction of a *smooth* coarse model F_H that would ensure a decrease of the fine level objective function F_h. We can extend these ideas to propose *non-smooth* coarse models with similar guarantees. A simple additional assumption is nonetheless required to ensure the decrease of the fine level objective function with such coarse level.

Assumption 2. ***Smoothing type.*** *Let $\gamma_H > 0$. Let R_H be a smoothable convex function (Definition 19), with an inf-conv γ_H-smooth approximation (Definition 20), where for all x $\omega(x) \geq 0$. This approximation $R_{H,\gamma_H} : \mathcal{H} \to (-\infty, +\infty]$ is such that there exist $\eta > 0$ such that the following hold:*

$$(\forall x \in X) \quad R_H(x) - \eta\gamma \leq R_{H,\gamma_H}(x) \leq R_H(x). \tag{4.38}$$

This assumption is met by inf-conv γ_H-smooth approximation (Definition 20) as soon as $\omega(\mathbf{0}) = 0$ (Lemma 7). It is the case of the Moreau envelope and of other choices of ω constructed with norms.

Definition 27. ***Non-smooth coarse model*** F_H**.** *The coarse model F_H is defined for the point $y_h \in \mathbb{R}^{N_h}$ as:*

$$F_H = L_H + R_H + \langle v_H, \cdot \rangle, \tag{4.39}$$

where

$$v_H = I_h^H \left(\nabla L_h(y_h) + \nabla R_{h,\gamma_h}(y_h)\right) - \left(\nabla L_H(I_h^H y_h) + \nabla R_{H,\gamma_H}(I_h^H y_h)\right).$$

Remark 10. *Two remarks. The coarse model is no longer constructed with a smooth regularization and thus the first order coherence is imposed between smoothed version of the fine level* and *of the coarse level objective functions.*

Definition 28. ***First order coherence between smoothed functions.*** *The first order coherence between the smoothed version of the objective function F_h at the fine level and the smoothed coarse level objective function F_H is verified in a neighborhood of y_h if the following equality holds:*

$$\nabla \left(L_H + R_{H,\gamma_H}\right)(I_h^H y_h) = I_h^H \nabla \left(L_h + R_{h,\gamma_h}\right)(y_h). \tag{4.40}$$

The defined coarse model obviously respects this definition.

Lemma 11. *If F_H is given by Definition 27, it necessarily verifies the first order coherence (Definition 28).*

Proof. Straightforward. □

Now it remains to show that decreasing this non-smooth coarse model will decrease the fine level objective function.

Fine model minimization with non-smooth multilevel steps. With the previous definitions of F_H, $\Phi_{H,\bullet}$ and I_h^H, the following lemmas prove that minimization at the coarse level also induces a descent direction at the fine level.

Lemma 12. ***Descent direction for the fine level smoothed function.*** *Let us assume that I_h^H and I_H^h are CIT operators and that F_H satisfies Definition 27 and Assumption 2 and that $\Phi_{H,\bullet}$ verifies Assumption 1. Then, $I_H^h(x_{H,m} - x_{H,0})$ is a descent direction for $L_h + R_{h,\gamma_h}$.*

Proof. Set $y_h \in \mathbb{R}^{N_h}$ and let us define $p_H := x_{H,m} - x_{H,0}$. Recall that $x_{H,0} = I_h^H y_h$. From Assumption 1 we have that $F_H(x_{H,m}) \leq F_H(x_{H,0})$, which is

$$L_H(x_{H,m}) + R_H(x_{H,m}) \leq L_H(x_{H,0}) + R_H(x_{H,0}) \tag{4.41}$$

By taking the inf-conv γ_H-smooth approximation of R_H we have that:

$$L_H(x_{H,m}) + R_{H,\gamma_H}(x_{H,m}) \leq L_H(x_{H,0}) + R_{H,\gamma_H}(x_{H,0}) \tag{4.42}$$

Thus:

$$\langle p_H, \nabla F_{H,\gamma_H}(x_{H,0})\rangle \leq 0.$$

By the first order coherence and imposing $I_h^H = \nu^{-1}\left(I_H^h\right)^T$ we obtain

$$\langle p_H, \nabla F_{H,\gamma_H}(x_{H,0})\rangle = \langle p_H, I_h^H \nabla(L_h + R_{h,\gamma_h})(y_h)\rangle = \nu^{-1}\langle I_H^h(p_H), \nabla(L_h + R_{h,\gamma_h})(y_h)\rangle \leq 0.$$

□

Now with a proper step size $\bar{\tau}_h$ that guarantees:

$$(L_h + R_{h,\gamma_h})(y_h + \bar{\tau}_h I_H^h(x_{H,m} - x_{H,0})) \leq (L_h + R_{h,\gamma_h})(y_h),$$

we recover the fine level decrease property:

Lemma 13. ***Fine level decrease.*** *If the assumptions of Lemma 12 hold, the iterations of Algorithm 4 ensure:*

$$F_h(y_h + \bar{\tau} I_H^h(x_{H,m} - x_{H,0})) \leq F_h(y_h) + (\eta_1 + \eta_2)\gamma_h. \tag{4.43}$$

This allows us to use a non-smooth coarse model at coarse level, which in case of an explicit proximity operator for R_h may be of interest as a non-smooth coarse model should approximate better the fine level objective function.

4.5.4 Asymptotic convergence guarantees

In order to obtain the convergence of the iterates to a minimizer of F_h and the optimal rate of convergence of the objective function values, we need to take into account two types of inexactness in the computation of one iterate: one on the proximity operator of R_h and one on the gradient of L_h. The error on the gradient will allow us to model coarse corrections to the fine level sequence with our multilevel framework, while the error on the proximity operator will allow us to consider approximation of proximity operators whose closed form is unknown.

The goal of this section is to show that an iteration of our algorithm (Steps 12-14 in Algorithm 4) can be reformulated as:

$$\begin{aligned} x_{h,k+1} &\approx_{i,\epsilon_{h,k}} \operatorname{prox}_{\tau_{h,k} R_h}\left(y_{h,k} - \tau_{h,k}\nabla L_h\left(y_{h,k}\right) + c_{h,k}\right), \\ y_{h,k+1} &= x_{h,k+1} + \alpha_{h,k}(x_{h,k+1} - x_{h,k}), \end{aligned} \tag{4.44}$$

where we introduce $c_{h,k}$ to model uncertainties on the gradient step due to the multilevel corrections and the pair $(i, \epsilon_{h,k})$ introduced in (4.26), to designate the type and the accuracy of the proximity operator approximation. Such rewriting allows us to fit in the framework described by the authors of [70] to define an inexact and inertial forward-backward algorithm.

Inexactness due to coarse corrections. As presented in the algorithm, a coarse correction is inserted before a typical fine level step. We can see the coarse correction as some kind of error on the gradient of L_h. In a typical multilevel step, at the fine level (cf. Steps 12 and 8 of Algorithm 4), the update would simply take the form:

$$\bar{y}_{h,k} = y_{h,k} + \bar{\tau}_{h,k} I_H^h (x_{H,k,m} - x_{H,k,0}), \tag{4.45}$$

$$x_{h,k+1} \approx_{i,\epsilon_{h,k}} \operatorname{prox}_{\tau_{h,k} R_h} \left(\bar{y}_{h,k} - \tau_{h,k} \nabla L_h \left(\bar{y}_{h,k} \right) \right), \tag{4.46}$$

$$y_{h,k+1} = x_{h,k+1} + \alpha_{h,k} (x_{h,k+1} - x_{h,k}). \tag{4.47}$$

It is easy to see that the coarse corrections are finite as we sum a finite number of bounded terms, thanks to computing updates at the coarse level with a Lipschitz gradient. This reasoning is detailed in the following proof for completeness of the argument.

Lemma 14. ***Coarse corrections are finite.*** *Let β_h and β_H be the Lipschitz constants of the gradients of L_h and L_H, respectively. Assume that we compute at most p coarse corrections. Let $\tau_{h,k}, \tau_{H,l} \in (0, +\infty)$ be the step sizes taken at fine and coarse levels, respectively. Assume that $\tau_{H,l} < \beta_H^{-1}$ and that $\tau_{h,k} < \beta_h^{-1}$ and denote $\bar{\tau}_h = \sup_k \bar{\tau}_{h,k}$. Then the sequence $(c_{h,k})_{k \in \mathbb{N}}$ in $\mathbb{R}^{N_h}$ generated by Algorithm 4 is defined as:*

$$c_{h,k} = \tau_{h,k} \left(\nabla L_h(y_{h,k}) - \nabla L_h(\bar{y}_{h,k}) + (\tau_{h,k})^{-1} \bar{\tau}_{h,k} I_H^h (x_{H,k,m} - x_{H,k,0}) \right), \tag{4.48}$$

if a coarse correction has been computed, and $c_{h,k} = 0$ otherwise. This sequence is such that $\sum_{k \in \mathbb{N}} k \|c_{h,k}\| < +\infty$.

Proof. $c_{h,k}$ only concerns the gradient update, so we focus on the forward step. Considering

$$\nabla L_h \left(\bar{y}_{h,k} \right) = \nabla L_h \left(\bar{y}_{h,k} \right) - \nabla L_h \left(y_{h,k} \right) + \nabla L_h \left(y_{h,k} \right),$$

and that we can rewrite $\bar{y}_{h,k} = \bar{y}_{h,k} + y_{h,k} - y_{h,k}$, the forward step can be rewritten as:

$$\begin{aligned} \bar{y}_{h,k} - \tau_{h,k} \nabla L_h(\bar{y}_{h,k}) = y_{h,k} - \tau_{h,k} \nabla L_h(y_{h,k}) \\ + \tau_{h,k} \left(\nabla L_h(y_{h,k}) - \nabla L_h(\bar{y}_{h,k}) + \frac{1}{\tau_{h,k}} (\bar{y}_{h,k} - y_{h,k}) \right). \end{aligned}$$

Therefore, each time a multilevel step is performed, it induces at iteration k, an error that reads:

$$c_{h,k} = \tau_{h,k} \left(\nabla L_h(y_{h,k}) - \nabla L_h(\bar{y}_{h,k}) + (\tau_{h,k})^{-1} \bar{\tau}_{h,k} I_H^h (x_{H,k,m} - x_{H,k,0}) \right).$$

Now, assuming that we use inertial inexact proximal gradient steps at the coarse level[3], the corresponding minimization verifies Assumption 1 on the decrease of F_H. It also produces bounded sequences if constructed according to the rules of [70, Definition 3.1, Theorem 4.1] as the sequences $(x_{H,k,\ell})_{k \in \mathbb{N}, \ell \in \mathbb{N}^*}$ converge. The sequence $(c_{h,k})_{k \in \mathbb{N}}$ has at

[3]The most general assumption in Assumption 1.

most p non-zero terms, that are bounded as shown below:

$$\tau_{h,k}^{-1}\|c_{h,k}\| = \|\nabla L_h(y_{h,k}) - \nabla L_h(\bar{y}_{h,k}) + (\tau_{h,k})^{-1}\bar{\tau}_h I_H^h(x_{H,k,m} - x_{H,k,0})\| \tag{4.49}$$

$$\leq \beta_h\bar{\tau}_h\|I_H^h(x_{H,k,m} - x_{H,k,0})\| + (\tau_{h,k})^{-1}\bar{\tau}_h\|I_H^h(x_{H,k,m} - x_{H,k,0})\| \tag{4.50}$$

$$\leq \bar{\tau}_h\left(\beta_h + \frac{1}{\tau_{h,k}}\right)\|I_H^h(x_{H,k,m} - x_{H,k,0})\|. \tag{4.51}$$

The second inequality is deduced from the fact that L_h has a β_h-Lipschitz gradient and that $\bar{y}_{h,k} - y_{h,k} = \bar{\tau}_{h,k} I_H^h(x_{H,k,m} - x_{H,k,0})$. Finally, as $(\|x_{H,k,0} - x_{H,k,m}\|)_{k\in\mathbb{N}}$ is bounded, we have:

$$\tau_{h,k}^{-1}\|c_{h,k}\| \leq \bar{\tau}_h\left(\beta_h + \frac{1}{\tau_{h,k}}\right)\sup_{k\in\mathbb{N}}\|I_H^h(x_{H,k,m} - x_{H,k,0})\| < +\infty. \tag{4.52}$$

□

Convergence of IML FISTA (Algorithm 4). We now discuss the convergence of our algorithm for the three types of approximation of the proximity operator introduced in Section 4.3.

We first consider a standard inexact forward-backward with a finite number of multi-level coarse corrections.

Theorem 4 (Approximation of Type 0)**.** *Let us suppose in Algorithm 4 that* $\forall k \in \mathbb{N}^*$, $\alpha_{h,k} = 0$ *at step 14, that the assumptions of Lemma 14 hold, and that the sequence* $(\epsilon_{h,k})_{k\in\mathbb{N}}$ *is such that* $\sum_{k\in\mathbb{N}}\sqrt{\|\epsilon_{h,k}\|} < +\infty$*. Set* $x_{h,0} \in \mathbb{R}^{N_h}$ *and choosing approximation of Type 0, the sequence* $(x_{h,k})_{k\in\mathbb{N}}$ *converges to a minimizer of* F_h*.*

Proof. The proof stems from Theorem 3.4 in [27] applied to the defined sequence. □

Theorem 5 (Approximations of Type 1 and Type 2)**.** *Let us suppose in Algorithm 4, that* $\forall k \in \mathbb{N}^*$, $t_{h,k+1} = \left(\frac{k+a}{a}\right)^d$*, with* (a, d) *satisfying the conditions in [70, Definition 3.1], and that the assumptions of Lemma 14 hold. Moreover, if we assume that:*

- $\sum_{k=1}^{+\infty} k^d\sqrt{\epsilon_{h,k}} < +\infty$ *in the case of Type 1 approximation,*
- $\sum_{k=1}^{+\infty} k^{2d}\epsilon_{h,k} < +\infty$ *in the case of Type 2 approximation,*

then, we have that:

- *The sequence* $(k^{2d}(F_h(x_{h,k}) - F_h(x^*)))_{k\in\mathbb{N}}$ *belongs to* $\ell_\infty(\mathbb{N})$*.*
- *The sequence* $(x_{h,k})_{k\in\mathbb{N}}$ *converges to a minimizer of* F_h*.*

Proof. [70, Theorem 3.5, 4.1, and Corollary 3.8] with Lemma 14 yield the desired result. □

Theorem 4 and 5 encompass convergence results obtained in [128, Theorem 1], [127, Theorem 1] and [126, Theorems 2.15, 2.16].

Remark 11. *The convergence results of Theorems 4 and 5 rely on quite weak assumptions. The advantage is that such assumptions encompasses a lot of previously proposed multilevel algorithms. We will not make of list of these algorithms here, but when discussing aspects and problems we have in common with them, we will refer to these two theorems.*

4.5.5 Extension to the multilevel case

In order to define more than two levels, it suffices to apply the same reasoning as in the two-level case but for the coarse level. Let us define a three-level algorithm. The fine level has a coarse model constructed according to Definition 25. This coarse level is smooth, therefore we can use directly the tools of smooth multilevel optimization (Chapter 2, Definition 15) to define the coarsest level, which is the coarse level of our coarse level. And so on if we want to define more levels.

As the construction we presented does not depend on the number of levels, the convergence results can be extended to more than two levels.

If the algorithm is used on J levels, we just have to apply the analysis derived above to each pair of consecutive levels. Then, recursively, showing that the coarsest level produces a bounded coarse correction will ensure that the upper finer level will converge to one of its minimizers, producing in turn a bounded coarse correction for the next upper finer level, and so on. We present the proof of convergence for a J-levels algorithm for only one type of approximation of the proximity operator, the proof being identical for the other cases.

Theorem 6 (Convergence of J-levels algorithm. Approximation of Type 0)**.** *Let us suppose in Algorithm 4 that $\forall k \in \mathbb{N}^*$, $\alpha_{h,k} = 0$ at step 14, that the assumptions of Lemma 14 hold for every pair of levels, and that the sequence $(\epsilon_{h,k})_{k\in\mathbb{N}}$ is such that $\sum_{k\in\mathbb{N}} \sqrt{\|\epsilon_{h,k}\|} < +\infty$. Set $x_{h,0} \in \mathbb{R}^{N_h}$ and choosing approximation of Type 0, the sequence $(x_{h,k})_{k\in\mathbb{N}}$ converges to a minimizer of F_h.*

Proof. Index the levels from $j = 1$ (the coarsest) to $j = J$ (the finest). The sequence at level $j = 1$ converges, and thus is bounded.

From Lemma 14, the coarse corrections applied to level $j = k$, $k \in \{2, \dots, J\}$ from level $j = k - 1$ are finite, assuming that the sequence at level $j = k - 1$ converges.

Therefore, by applying Theorem 4, the sequence generated by level $j = k$ converges to a minimizer of the associated problem.
By induction, the sequence generated by level $j = J$ converges to a minimizer of F_h. □

4.5.6 When to use the coarse models

We discuss now the decision rule to use the multilevel hierarchy. We want to answer the following question:

When should we use coarse corrections in the multilevel algorithm ?

The most common attempt to answer this question in multilevel optimization literature is derived from trust-region methods (see for instance [98, 102] and [89, 90, 96]) and use

one or combine some of the three conditions below:

$$\begin{aligned} \|I_h^H \nabla F_h(x_{h,k})\| &> \kappa \|\nabla F_h(x_{h,k})\|, \\ \|I_h^H \nabla F_h(x_{h,k})\| &> \xi, \\ \|x_{h,k} - \tilde{x}\| &> \vartheta \|\tilde{x}\|, \end{aligned} \tag{4.53}$$

where $\tilde{x}$ is the last iterate that received a coarse correction, and κ, ξ, ϑ are positive parameters. The intention behind checking these conditions is to:

(i) only use coarse corrections when the first coarse step is relatively big compared to the fine level step;

(ii) verify if the first iterate at coarse level is not too close to the optimum of F_H, which, according to the literature [89, 90], would make a coarse correction less impactful;

(iii) compute coarse corrections after some progress has been made at fine level.

In a vacuum these three conditions seem reasonable, but they have several limits:

- If one uses first order methods, computing at each iteration a restriction at coarse level of the gradient even when one does not use coarse corrections[4] is costly: one needs to compute a useless matrix-vector product in the fine level space.
- As we only force local first order coherence, it may be useful to go back at fine level after a few coarse iterations to update this coherence and then go down again. Such chain of coarse corrections iterations may lead to faster convergence (see section 5) and reduced computation time as a result.

Multilevel methods are of interest to reduce computation time, using these conditions instead of setting when to use the coarse corrections before the optimization could hurt our efforts. Nevertheless, they are interesting for high order multilevel optimization where computing a restriction of the gradient is almost negligible compared to other computations [102]. We choose to drop such conditions in our experiments and choose a fix predefined multilevel pattern as in classical multigrid. Such pattern can be identified experimentally, and we will discuss this in Chapter 5.

4.6 Concurrent frameworks

In this section we present concurrent frameworks proposed in the literature to define multilevel proximal algorithms, and to highlight their strengths and key differences with our approach. As previously stated, we have identified three of them [88–90] that we will discuss in chronological order of publication. All three of these algorithms are designed to tackle composite optimization problems of the form of Problem (2.7): the sum of one continuously differentiable function and one non-smooth function. None of the three proposed algorithms allow for inexactness in the proximity operator, so we won't emphasize this aspect in the following, but it should be kept in mind. For the presentation, we used our notations, instead of the notations of the respective articles, to avoid confusion.

[4] There is not a lot of them used in practice (see for instance the experiments done in [89, 90, 108] or our own in Chapter 5)

MAGMA: Multilevel Accelerated Gradient Mirror Descent Algorithm [89]. Parpas and co-authors have proposed in [89] a multilevel algorithm based on gradient and mirror descent. The idea behind this choice is the interpretation of Nesterov acceleration in [148] as a linear coupling between gradient and mirror descent steps[5].

At each iteration MAGMA performs both gradient and mirror descent steps, then uses a convex combination of their results to compute the next iterate. When a coarse correction is computed, this correction replaces the proximal gradient step.

The construction of the coarse model is similar to ours in the sense that it is chosen smooth with the first order coherence enforced with the smoothed fine level objective function (using the framework of [132]). Step (14) of MAGMA is the Mirror descent step

Algorithm 5 MAGMA [89]

1: Set $x_{h,0}, y_{h,0} \in \mathbb{R}^N$, $t_{h,0} = 1$
2: **while** $k = 0, 1, 2, \ldots, T$ **do**
3: Choose δ_{k+1} and α_{k+1} [89, Eq. 3.18,3.19].
4: Set $t_k = \frac{1}{\alpha_{k+1}\delta_{k+1}}$.
5: Set $x_k = t_k z_k + (1 - t_k) y_k$.
6: **if** Conditions (4.53) are satisfied **then**
7: $x_{H,k,0} = I_h^H y_k$
8: Compute m gradient steps on F_H.
9: Set $\bar{\tau}_k > 0$ [89, Eq. 3.10],
10: $y_{k+1} = y_k + \bar{\tau}_k I_H^h \left(x_{H,k,m} - x_{H,k,0}\right)$
11: **else**
12: $y_{k+1} = \text{prox}_{1/\beta_h R_h}\left(y_k - 1/\beta_h \nabla L_h\left(y_k\right)\right)$
13: **end if**
14: Set $z_{k+1} = \text{Mirror}_{z_k}(\nabla L_h(x_k), \alpha_{k+1})$
15: **end while**

on L defined by [89, Definition 2.2], where α_{k+1} is the associated step size. It is assumed to be available in closed form. With the right choice of underlying potential for the Mirror descent step, one can recover the proximal gradient step.

In all their experiments, for the Mirror operator authors of [89] chose the standard Euclidean norm $\|\cdot\|_2$ and accordingly $\frac{1}{2}\|\cdot\|_2^2$ as a Bregman divergence, which transforms the Mirror step in a proximal gradient step. In this setting, each coarse correction step is followed by a proximal gradient step (Step (14)), but a key difference with our method is that the Mirror step *does not take into account the coarse correction*: the Mirror step is computed with the previous iterate instead.

Another key difference is that *the cost per iteration when coarse models are unused is at least twice the cost of our method*: one proximal gradient step and one mirror step. When using coarse corrections, the cost is the same as our method: one coarse correction step and one mirror step (instead of a proximal gradient step in IML FISTA).

The convergence of the algorithm MAGMA was shown with respect to objective func-

[5]To keep the presentation as simple as possible we do not detail the mirror descent step in the general case, see [89, 148].

tion values assuming that the smoothing parameter is chosen according to the relationship

$$0 < \gamma_k \leq \frac{\xi}{(\beta_h + \delta_{k+1})\alpha_{k+1}\eta T} \tag{4.54}$$

for a small $\xi \in (0, 1)$, objective function values generated by MAGMA converge with a rate $1/k^2$ to the minimum of Problem (2.7). This forces the smoothing parameter to be quite small in practice. In our own experiments we observed that small smoothing parameters led to small coarse correction steps at the fine level, and thus slower convergence. This choice of smoothing parameter is also necessary to prove the convergence of MPGM which is discussed next.

A summary of the differences between our work and what was proposed in [89] is given below:

1. The smoothing parameter cannot be tuned as freely as in our method, and it may slow down the convergence in practice.

2. The use of non-smooth coarse models was not considered, only smooth ones are permitted.

3. Same convergence rate as ours, but a complexity per iteration potentially twice as high: this ascertains our choice of FISTA as a fine level algorithm.

4. The convergence of the sequence of iterates to a minimizer was not discussed.

MPGM: Multilevel Proximal Gradient Method [90]. It was later proposed by the same author in [90] to define a multilevel algorithm, restricted to proximal gradient updates at fine level this time. This algorithm was used for one image restoration task: deblurring regularized with an ℓ_1-wavelet penalty, where MPGM showed great convergence speed in practice. MPGM was proposed to solve problems where L was possibly non-convex, and consequently L_H could also be non-convex. Therefore, a line search is added at each step at coarse level to guarantee that the coarse correction will be a descent direction for the smoothed fine level objective function (Lemma 9). This line-search was first proposed in [103], specifically for multigrid algorithm applied on non-convex optimization. It is not necessary to use it in the convex case as the first condition

$$F_H(x_{H,k,0}) + \kappa_2 \langle \nabla F_H(x_{H,k,0}), x_{H,k,\ell+1} - x_{H,k,0} \rangle < F_H(x_{H,k,\ell+1}) \tag{4.55}$$

directly comes from the convexity of F_H if $F_H(x_{H,k,\ell+1}) < F_H(x_{H,k,0})$. Thus, the coarse model minimization is similar to the one used in MAGMA and our method in the convex case. Again, similarly to MAGMA, the smoothing parameter is reduced at each iteration to guarantee convergence of the algorithm.

If the framework shares similarities with our own method, there are notable differences in the choice of coarse model in the applications. The choice of smoothing method for the ℓ_1-norm is suboptimal in the experiments done in [90]. Indeed, the regularization is smoothed using $x \mapsto \sum_{i=1}^{N} \sqrt{x_i^2 + \gamma}$, which is a factor 2 worse than the Moreau envelope (see end of Section 4.2.2). Moreover, the information transfer operator used is the most standard one (Equation (3.12)). Finally, in the reported results, only 20 iterations of

Algorithm 6 MPGM

Set $x_{h,0} \in \mathbb{R}^N$,
Line search parameters: $0 < \kappa_1 < \frac{1}{2}$, $1 - \kappa_1 \leq \kappa_2 \leq 1$
$\gamma > 0$ and $0 < \delta < 1$, and $r = 0$.
while Stopping criterion is not met **do**
  if Conditions (4.53) are satisfied **then**
    $x_{H,k,0} = I_h^H x_{h,k}$
    Set $\gamma_k = \gamma\delta^r$, Smoothing adjustment,
    for $\ell = 0, 1, \ldots, m-1$ **do**
      Set $\tau_{H,l} > 0$ such that
      $F_H(x_{H,k,0}) + \kappa_2 \langle \nabla F_H(x_{H,k,0}), x_{H,k,\ell+1} - x_{H,k,0} \rangle < F_H(x_{H,k,\ell+1})$
      and $F_H(x_{H,k,\ell+1}) \leq F_H(x_{H,k,\ell}) - \kappa_1 \tau_{H,l} \|\nabla F_H(x_{H,k,\ell})\|^2$
      $x_{H,k,\ell+1} = x_{H,k,\ell} - \tau_{H,l} \nabla F_H(x_{H,k,\ell})$
    end for
    Set $\bar{\tau}_{h,k} > 0$,
    $\bar{x}_{h,k} = x_{h,k} + \bar{\tau}_{h,k} I_H^h \left(x_{H,k,m} - x_{H,k,0}\right)$
    $r = r + 1$
  else
    $\bar{x}_{h,k} = x_{h,k}$
  end if
  $x_{h,k+1} = \text{prox}_{1/\beta_h R_h} \left(\bar{x}_{h,k} - 1/\beta_h \nabla L_h \left(\bar{x}_{h,k}\right)\right)$
end while

MPGM, FB and FISTA were displayed, which is a bit small to compare the convergence speed of the three algorithms. FISTA may to be slower than FB in the first iterations in some contexts.

A summary of the differences between our work and what was proposed in [90] is given below:

1. The smoothing parameter cannot be tuned as freely as in our method, and it may slow down the convergence in practice.

2. The use of non-smooth coarse models was not considered.

3. Global convergence of the algorithm in the case of non-convex function L (R was still assumed convex) is shown (the minimum of all the steps taken goes to 0 when k goes to infinity) [90, Theorem 3.1].

4. A convergence rate was specified for strongly convex functions.

5. Extrapolation steps were not used, thus a slower convergence rate is expected.

6. The convergence of the sequence of iterates to a minimizer was not discussed.

If we implement rigorously MPGM to compare it to our algorithm, it is expected to be slower than IML FISTA due to the fact that a condition that computes the gradient of the smoothed fine level objective function, and its projection to the coarse level, is checked at each iteration. This is a costly operation, that is not particularly useful in the context

of our imaging applications as we will see it in Chapters 5 and 6. Therefore, such a comparison would be unfair to MPGM.

MGProx: MultiGrid Proximal gradient method [88]. In a recent paper, Ang and co-authors [88] proposed a framework to define multilevel algorithms for non-smooth optimization. The presented construction is mostly suited for strongly convex objective function, and in the rest of this paragraph we will make this assumption unless stated otherwise.

Their main idea is to define the first order coherence using a subgradient selection (as opposed to the smoothing). This subgradient selection is done with what is called adaptive information transfer operators. We reproduce the definition they provided here:

Definition 29. ***Adaptative restriction operator for separable regularization R [88, Definition 2.3].*** *For a possibly non-smooth function $R : \mathbb{R}^N \to \mathbb{R}$ that is separable, i.e. $R(x) = \sum_{i=1}^{N} R_i(x_i)$, given a full restriction operator I_h^H and a vector x, the adaptive restriction operator $\bar{I}_h^H$ with respect to R at x is defined by zeroing out the columns of I_h^H corresponding to the elements in ∂R that are set-valued.*

As the (Minkowski) sum of the set of subgradients of two functions is not always equal to the subgradients of the sum of those two functions (Moreau-Rockafellar theorem [149]), the adaptive restriction operator removes any possible ambiguity. The first order coherence is then defined as choosing an element of the set

$$\partial F_H(x_{H,k,0}) - \bar{I}_h^H \partial F_h(y_{h,k+1}) \tag{4.56}$$

where the right-hand side $\bar{I}_h^H \partial F_h(y_{h,k+1})$ is uniquely valued. After *minimization of the coarse model* to obtain $x_{h,k+1}$, the coarse correction sent to the fine level respects the following strict inequality ($\leq$ if F is only convex) [88, Theorem 2.5]:

$$\langle \partial F_h(y_{h,k+1}), I_H^h(x_{H,k+1} - x_{H,k,0}) \rangle < 0, \tag{4.57}$$

i.e. every element of $\partial F_h(y_{h,k+1})$ is negatively correlated with $I_H^h(x_{H,k+1} - x_{H,k,0})$. Now to show that this constitutes a descent direction for the fine level function, we need the following notions:

Definition 30. ***Support function of a convex set [51].*** *The support function $\sigma_C : \mathbb{R}^n \to \mathbb{R}$ of a non-empty closed convex set C in $\mathbb{R}^n$ is given by:*

$$\sigma_C(x) = \sup_{s \in C} \langle s, x \rangle \tag{4.58}$$

The subdifferential of a convex function at a point x can be characterized equivalently by the support function of its directional derivative (Subdifferential I), or by sublinearity property of the function (Subdifferential II). The former is useful to characterize descent directions and steepest descent direction, which is precisely what was used by the authors of [88].

Definition 31. ***Subdifferential and subgradient of convex functions [51].*** *Let g be a proper, lower semi-continuous, convex function on $\mathbb{R}^n$. The subdifferential of g is defined by the following sets:*
Subdifferential I. *The subdifferential $\partial F(x)$ of F at x is the nonempty compact convex set of $\mathbb{R}^n$ whose support function is $F'(x,\cdot)$, i.e.*

$$\partial F(x) := \{s \in \mathbb{R}^n | \langle s, d\rangle \leq F'(x,d) \text{ for all } d \in \mathbb{R}^n\} \tag{4.59}$$

We have the following relationship that links the support function to the directional derivative (if $F(x)$ is finite):

$$F'(x;d) = \sigma_{\partial F(x)} = \max\left\{\langle s, d\rangle, s \in \partial F(x)\right\},$$

thus

$$F_h'(y_{h,k+1}; I_H^h(x_{H,k+1} - x_{H,k,0})) = \max\left\{\langle s, I_H^h(x_{H,k+1} - x_{H,k,0})\rangle, s \in \partial F_h(y_{h,k+1})\right\} < 0.$$

This proves that there exists $\bar{\tau}_{h,k} > 0$ such that [88, Lemma 2.7]:

$$F_h(y_{h,k+1} + \bar{\tau}_{h,k} I_H^h\left(x_{H,k,m} - x_{H,k,0}\right)) < F_h(y_{h,k+1}).$$

However, there is no available line-search that is guaranteed to find such a $\bar{\tau}_{h,k}$ in practice. Moreover, if F_h is only convex, $\bar{\tau}_{h,k}$ is not guaranteed to be strictly positive. The conver-

Algorithm 7 MGProx

Set $x_{h,0} \in \mathbb{R}^N$,
while Stopping criterion is not met **do**
  $y_{h,k+1} = \text{prox}_{1/\beta_h R_h}\left(\bar{x}_{h,k} - 1/\beta_h \nabla L_h\left(\bar{x}_{h,k}\right)\right)$
  Construct the adaptive restriction operator I_h^H
  $x_{H,k,0} = I_h^H y_{h,k+1}$ Projection
  Select a subgradient g_{k+1} of $\partial F_H(x_{H,k,0}) - I_h^H \partial F_h(y_{h,k+1})$
  $x_{H,k+1} = \arg\min_x F_H(x) - \langle g_{k+1}, x\rangle$
  Set $\bar{\tau}_{h,k} > 0$,
  $\bar{y}_{h,k+1} = y_{h,k+1} + \bar{\tau}_{h,k} I_H^h\left(x_{H,k,m} - x_{H,k,0}\right)$
  $x_{h,k+1} = \text{prox}_{1/\beta_h R_h}\left(\bar{y}_{h,k+1} - 1/\beta_h \nabla L_h\left(\bar{y}_{h,k+1}\right)\right)$
end while

gence of the algorithm MGProx was shown with respect to objective function values. The authors of [88] showed that the objective function values generated by MGProx converge to the minimum of Problem (2.7), with a rate $1/k$ or $1/k^2$ if inertia was used.

The use of adaptive restriction operators is a great idea to define the first order coherence, however it limits the application of the framework to separable regularizations. Moreover, a choice of subgradient must be made each time a coarse model is used, which may be difficult to do correctly in practice. The authors of [88] chose to put to zero the set elements of the subgradient, which works fine for the ℓ_1-norm whose subgradient at 0 contains 0. However, for a convex function that does not have such property, zero-ing

would not work. Also, the descent guarantee coming back from the coarse level are equivalent to ours when F_h is only convex and not strongly convex, which is the case in a lot of applications. Indeed, there is no guarantee to find a positive step size that ensures decrease of the fine level (Steps (8), (9) in MGProx).

MGProx was applied only on a PDE problem, which is a really favorable context for multilevel algorithms (recall our discussions in Chapter 3). Therefore, the results are quite impressive. A summary of the differences between our work and what was proposed in [88] is given below:

1. The first order coherence is more rigorously defined than ours for non-smooth optimization but only applicable when the problem is separable.

2. Coarse corrections are fixed points of the proximal gradient steps at fine level, a property that is not always preserved when using smoothing.

3. The coarse model needs to be minimized (versus decreased in our method) to guarantee a descent direction.

4. The cost per iteration is at least twice as high as ours: two proximal gradient step are used at the fine level, and the coarse level needs to be minimized.

5. The convergence of the sequence of iterates to a minimizer was not discussed.

In the end, one could plug the construction of coarse model developed in [88] in our framework to obtain a more general multilevel algorithm. Due to the late discovery of this work, we leave to later work the practical comparison of the two methods to define coarse models. In the next section, we will also argue that there are equivalences to be drawn when dissecting the smoothing framework.

IML FISTA versus MPGM, MAGMA and MGProx. We summarize the main differences of the three algorithms presented above with our proposed IML FISTA in Table 4.1.

4.7 Conclusion

In this chapter, we laid the theoretical foundations for our algorithm IML FISTA. We recovered optimal convergence guarantees of the literature on the class of functions we consider in this manuscript. However, the proposed construction does not answer the most important question of multilevel algorithm: will it perform better than its single level counterpart? There does not exist a theoretical (and definite) answer in general, thus we will rely on numerical experiments in the next chapters to confirm the relevance of our approach.

	Guarantees			
	Generality	Inexactness	$\left(F(x_k) - F^*\right) \leq O(1/k^2)$	$x_k \to x^*$
MAGMA [89]	✓	✗	✓	✗
MPGM [90]	✓	✗	✗	✗
MGProx [88]	✗	✗	✓	✗
IML FISTA	✓	✓	✓	✓

Table 4.1: In this table, we summarize the guarantees of the multilevel algorithms presented in this section with respect to our proposed IML FISTA. The first column of the guarantees (Generality) indicates whether the algorithm is applicable on all problem of the form (2.7) (i.e. when F is convex, the proximity operator of R is available and L has a $\beta-$Lipschitz gradient), while the second column refers to the inexactness of the proximity operator, the third to the convergence rate of objective function values, and the fourth to the convergence to a minimizer.

Chapter 5

IML FISTA: applications to image restoration

In this chapter we present the three applications of our algorithm IML FISTA on color and hyperspectral images. We start by demonstrating its potential on the reconstruction of color images. With this presentation comes a detailed discussion about the tuning of our algorithm's hyperparameters. Then, we will consider the reconstruction of hyperspectral images, to display IML FISTA's performance in a high dimensional context.

The content of this chapter was partially published in the following papers [126–128, 150]. The specific design of the multilevel algorithms was refined from [128] (submitted in March 2022) to [126] (submitted in July 2023), therefore this Chapter is a more coherent presentation of the experiments we conducted. We also added some unpublished experiments on low dimensional images to support conclusions made in [126]. The code to reproduce the experiments is available here: https://github.com/laugaguillaume/.

5.1 Introduction

We have seen in the previous chapter the design of a general framework that allows us to tackle all sort of optimization problems. With this generality also comes the lack of theoretical proof that our multilevel algorithm will perform better than its single level counterpart. This is a common issue in papers presenting multilevel optimization methods, and often they chose to demonstrate the efficiency of their algorithm on problems with a limited scope. We aim to enlarge this scope, to show that our algorithm can be applied to a wide range of problems, and where it may fail to outperform single level algorithms. With our experiments we want to identify as precisely as possible in which contexts our algorithm, and other multilevel algorithms should be considered (and subsequently when they should not).

The first question that comes to mind when trying to identify such contexts is the choice of hyperparameters, that are plenty. We list here the hyperparameters of our algorithm:

(i) the number of levels L,

(ii) the information transfer operators,

(iii) the functions at each level: L_H and R_H,

(iv) the choice of smoothing to define the first order coherence,

(v) the smoothing parameter γ,

(vi) the number of uses or calls to the coarse models p,

(vii) the path through the levels, i.e., in which order we go through the levels (e.g. V-cycle, W-cycle etc. [84, 86, 96]),

(viii) the minimization strategy at coarse level Φ_H,

(ix) the number of iterations at each level m,

We did not test exhaustively hyperparameters (i), (iii) and (vii) of this list. For the number of levels, we have seen in our experiments that $L = 5$ was a good choice, and that one would be even happy with $L = 2$ in all cases where multilevel algorithms work, i.e., when they outperform their single level counterpart. For the choice of coarse models, a lot of the possible functions are ineffective,for obvious reasons. We tested some of them, and it showed that going along the route of reduced order approximation is a good (and more importantly simple) choice in all configurations. Finally, we always use V-cycles in our experiments, as they are the simplest cycles, and the most common in the literature.

Organization of the chapter. In the first section of this chapter, we quickly present the fine level optimization problem that we will consider. From this definition, we specify the construction of the coarse model and information transfer operators, and then conduct a benchmark of the impact of the other hyperparameters. This benchmark will be used as a guideline for the rest of the applications considered in this chapter.

Then we specify IML FISTA to three different image restoration problems: image deblurring, image inpainting and then hyperspectral image reconstruction where both types of degradation are considered. In these sections, we will investigate the impact of the dimension on IML FISTA's performance, and we will see that it varies quite a lot depending on the problem. We conducted extensive experiments to show that IML FISTA, when compared to FISTA, is a good choice in terms of convergence and image quality, and that the chosen hyperparameters are robust across all problems.

5.2 Image restoration problems: data fidelity and regularization

For the rest of this chapter, let us specify Problem (2.7) to the specific context of image restoration in multilevel notations:

$$\widehat{x} \in \underset{x_h \in \mathbb{R}^{N_h}}{\operatorname{Argmin}} F_h(x) := f_h(\mathrm{A}_h x_h) + g_h(\mathrm{D}_h x_h) \tag{5.1}$$

with $A_h \in \mathbb{R}^{M_h \times N_h}$ and $\mathrm{D}_h \in \mathbb{R}^{(N_h \times \widetilde{K}) \times N_h}$ $(\widetilde{K}, M_h > 0)$. Recall that the inverse problem modeling is as follows, with an additive Gaussian noise ϵ:

$$z_h = \mathrm{A}_h \bar{x}_h + \epsilon.$$

The parameter $\tilde{K}$ expresses the fact that operator D_h can map x_h to a higher dimensional space, e.g. $\widetilde{K} = 2$ for Total Variation penalization. In this expression, $x_h = (x_h^i)_{1 \leq i \leq N_h}$ is the vectorized version of an image X_h of $N_{h,r}$ rows and $N_{h,c}$ columns, and where each pixel corresponds to a vector of $C \geq 1$ components (e.g. $C = 3$ for the RGB bands of a color image). Hence, we have $N_h = N_{h,r} \times N_{h,c} \times C$. In the following, as the operators we deal with are applied separately to each channel, for the sake of clarity and without loss of generality, we present their construction for grayscale images corresponding to $C = 1$. For the reader familiar with deblurring and inpainting problem, and total variation regularization, this section may be skimmed.

5.2.1 Data fidelity terms $f_h \circ \mathrm{A}_h$

We will consider two different types of degradation in this chapter: image blurring and image inpainting. Only for hyperspectral images, will we consider a combination of both. For these two types of degradation, we specify the construction of A_h and f_h.

Deblurring problem. When the degradation of the image corresponds to a blurring effect, the operator A_h is a convolution matrix built from a two-dimensional Point Spread Function (PSF). As it is the case for Gaussian blurs, the PSF function often takes the form of a separable kernel (horizontally and vertically) and A_h can be decomposed into a Kronecker product:

$$\mathrm{A}_h = \mathrm{A}_{h,r} \otimes \mathrm{A}_{h,c} \tag{5.2}$$

with $\mathrm{A}_{h,r} \in \mathbb{R}^{N_{h,c} \times N_{h,c}}$ and $\mathrm{A}_{h,c} \in \mathbb{R}^{N_{h,r} \times N_{h,r}}$ (r stands for row, c for columns). From the numerical viewpoint, this Kronecker decomposition is particularly efficient for processing large images, and can be easily implemented with the HNO package [3]. Finally, as it is common in image restoration, the data-fidelity term is the least square regression:

$$(\forall x_h \in \mathbb{R}^{N_h}) \qquad f_h(\mathrm{A}_h x_h) = \frac{1}{2}\|\mathrm{A}_h x_h - z_h\|_2^2 = \frac{1}{2}\sum_{i=1}^{N_h}((\mathrm{A}_h x_h)^i - (z_h)^i)^2. \tag{5.3}$$

Inpainting problem. When the degraded image coincides with the original image but with potentially altered or missing pixels, the reconstruction task is called inpainting and A_h is a measurement operator that keeps a subset $I \subseteq \{1, \ldots, N_h\}$ of pixels of the image and removes the others. Here, we assume that the subset I is chosen randomly. Formally A_h takes the form of a diagonal matrix with a Bernoulli random variable (zeros and ones) on its entries, and it plays the role of a mask applied to the image x_h:

$$(\mathrm{A}_h x_h)^i = \begin{cases} x_h^i & \text{if } i \in I \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

In this case too, the data-fidelity term is the least square regression as in Equation (5.3). Such inpainting problem is quite simple in its formulation, and far from the complexity of inpainting problems treated by deep learning techniques nowadays [151]. It is however still relevant for hyperspectral images, where similar degradation can occur [152].

5.2.2 Regularization terms $g_h \circ \mathrm{D}_h$

In our experiments we considered three types of regularization terms:

Wavelet transform norm. The operator D_h associated with a wavelet transform regularization is the discrete wavelet transform operator which computes a given number of consecutive decimated low pass and high pass filtering of the image x_h. The classical regularization associated is the application of the l_1-norm on the discrete wavelet transform coefficients.

Total Variation. The operator D_h associated with the Total Variation (TV) computes the first order differences between the component i of x_h and its horizontal/vertical nearest neighbors $(x_h^{i_c}, x_h^{i_r})$ (lower/right in the image case). It is defined such that for all $x_h \in \mathbb{R}^{N_h}$, and for each pixel $i \in \{1, \ldots, N_h\}$,

$$(\mathrm{D}_h x_h)^i = \left[\; x_h^i - x_h^{i_r} \, , \, x_h^i - x_h^{i_c} \; \right], \tag{5.5}$$

paying particular attention to the management of border effects. Here $\mathrm{D}_h x_h$ belongs to $\mathbb{R}^{N_h \times 2}$ ($\widetilde{K} = 2$). With this definition, the classical isotropic Total Variation semi-norm [57] reads:

$$g_h(\mathrm{D}_h x_h) = \lambda_h \sum_{i=1}^{N_h} \| \, (\mathrm{D}_h x_h)^i \, \|_2 = \lambda_h \sum_{i=1}^{N_h} \sqrt{|x_h^i - x_h^{i_1}|^2 + |x_h^i - x_h^{i_2}|^2} = \lambda_h \|\mathrm{D}_h x_h\|_{2,1} \tag{5.6}$$

with $\lambda_h > 0$.

Non-Local Total Variation. The operator D_h associated with the Non-Local Total Variation (NLTV) extends TV to a non-local neighborhood of the current pixel i. In words, it is the operator that computes the weighted differences between the current pixel i of an image x_h and a subset $\mathcal{N}_i$ of pixels localized near i.

For every $x_h \in \mathbb{R}^{N_h}$, and at each pixel $i \in \{1, \ldots, N_h\}$, for some given weights $\omega^{i,j} > 0$,

$$(\mathrm{D}_h x_h)^i = \left[\; \omega^{i,j} \left(x_h^i - x_h^j \right) \; \right]_{j \in \mathcal{N}_i}. \tag{5.7}$$

Here $\mathrm{D}_h x_h$ belongs to $\mathbb{R}^{N_h \times \widetilde{K}}$ and $\widetilde{K}$ is the cardinality of the subset $\mathcal{N}_i$. For every $i \in \{1, \ldots, N_h\}$ and $j \in \mathcal{N}_i$, the weights $\omega^{i,j} > 0$ depend on the similarity (e.g., ℓ_2 norm) between patches that are centered around components i and j of the image [33].

As for the isotropic TV semi-norm, an ℓ_p ($p \geq 1$) based NLTV semi-norm takes the form:

$$g_h(\mathrm{D}_h x_h) = \lambda_h \sum_{i=1}^{N_h} \| \, (\mathrm{D}_h x_h)^i \, \|_p \qquad \text{with} \qquad \lambda_h > 0. \tag{5.8}$$

5.3 Construction of the coarse models and the information transfer operators

In this section we adapt our Inexact MultiLevel FISTA to image reconstruction problems in the framework of Problem (5.1). We present our problem in a multilevel context, then

we propose CIT[1] operators designed for image reconstruction problems, and we derive the construction of a good coarse model through a specific choice of smoothing.

5.3.1 Information transfer operators

In the context of image reconstruction problems, we consider CIT operators that rely on wavelet bases (referred to as wavelet CIT in the following). The idea of constructing such information transfer operators traces back to works dedicated to image deblurring problems either based on biorthogonal wavelets [153] or Haar and Symlet wavelets [113, 114, 154]. Our objective is to obtain a computationally efficient coarse approximation of a vector lying in a higher resolution space, from the approximation coefficients of its discrete wavelet transform (DWT). We impose in this context that $N_h = (N_{h,r} \times N_{h,c}) = (2\,N_{H,r} \times 2\,N_{H,c}) = 4 \times N_H$. For a generic quadrature mirror filter $\mathbf{q} = (q_1, \ldots, q_m)$:

$$I_h^H := (\mathbf{R}_{\mathbf{q},r} \otimes \mathbf{R}_{\mathbf{q},c})\,, \tag{5.9}$$

where $\mathbf{R}_{\mathbf{q},c}$ is the decimated $N_{H,r}$-by-$N_{h,r}$ matrix (every other line is kept) of the $N_{h,r}$-by-$N_{h,r}$ Toeplitz matrix generated by $\mathbf{q}$ as :

$$\begin{pmatrix} q_1 & q_2 & \ldots & q_m & 0 & \ldots & 0 \\ 0 & 0 & q_1 & q_2 & \ldots & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ 0 & \ldots & 0 & 0 & 0 & q_1 & q_2 \end{pmatrix}.$$

Similarly, $\mathbf{R}_{\mathbf{q},r}$ is the decimated $N_{H,c}$-by-$N_{h,c}$ matrix (every other line is kept) of the $N_{h,c}$-by-$N_{h,c}$ Toeplitz matrix generated by $\mathbf{q}$. For both matrices the vector $\mathbf{q}$ is completed with the right number of 0's to reach the size $N_{h,r}$ or $N_{h,c}$. I_H^h is then taken in order to satisfy Definition 14. This Kronecker product structure is particularly interesting to accelerate the projection of fine level information to the coarse level as:

$$I_h^H x_h = \mathbf{R}_{\mathbf{q},c} X_h \mathbf{R}_{\mathbf{q},r}^T, \tag{5.10}$$

where X_h is the reshaped version of x_h in a matrix of size $N_{h,r} \times N_{h,c}$.

5.3.2 Fast coarse models

A challenging numerical problem is to keep the efficiency of matrix-vector product computation at coarse level if it exists at fine level. For instance, when considering convolutions, if the convolution matrix is expressed with a Kronecker product, such structure should be preserved with the right definition of operators at coarse levels.

A_H in the deblurring problem. Thanks to the Kronecker factorization of both A_h and I_h^H, the coarsened operator A_H can be written as:

$$\mathrm{A}_H = \left(\mathbf{R}_{\mathbf{q},c} A_{h,r} \mathbf{R}_{\mathbf{q},c}^T\right) \otimes \left(\mathbf{R}_{\mathbf{q},r} A_{h,c} \mathbf{R}_{\mathbf{q},r}^T\right)$$

[1] Coherent Information Transfer operators (see Chapter 4, Definition 14)

preserving the same computational efficiency. Thus, in image restoration problems where a separable blur is used, it is straightforward to design coarse operators (which can be computed beforehand) that are fast for matrix-vector products while keeping fidelity to the fine level.

A_H in the inpainting problem. Due to the specific diagonal form of A_h, the coarsened inpainting operator A_H simply stems from decimating the rows and the columns of A_h by a factor 2. $\mathrm{A}_H \in \mathbb{R}^{N_H \times N_H}$ remains a diagonal indicator matrix of a pixel subset $J \subseteq \{1, \dots, N_H\}$ acting as a mask on the coarse image:

$$(\mathrm{A}_H x_H)^j = \begin{cases} x_H^j & \text{if } j \in J \\ 0 & \text{otherwise} \end{cases}$$

Examples of operators D_H. For the regularization operators, the construction is simpler. Consider the case of the wavelet transform, the operator D_H is the decomposition with one level less. For both TV and NLTV, we use the same hyperparameters (maximum number of patches, size of patches, computation of similarity between patches, etc.) for D_H as for D_h. Adapting these parameters to current resolution could be worth investigating. However, due to the limited size of the chosen patches, we believe it would lead to marginal improvements. D_H is thus playing the same role as D_h but for images of size N_H. Here $\mathrm{D}_H x_H$ belongs to $\mathbb{R}^{N_H \times \widetilde{K}}$.

5.3.3 Choice of smoothing

A complete presentation of the coarse models cannot omit the choice of the smoothing technique to define the first order coherence between the fine and coarse levels. In this thesis, we chose to use the Moreau envelope in all of our experiments, with some little tweaks. The motivation is quite easy to understand: the Moreau envelope and the proximity operator are two sides of the same coin. In addition, the gradient of the Moreau envelope is directly expressed through the proximity operator (Chapter 4, Proposition 3):

$$(\forall x \in \mathcal{H}), \quad \nabla^\gamma g(x) = \frac{1}{\gamma}\left(\mathrm{Id} - \mathrm{prox}_{\gamma g}(x)\right).$$

Therefore, the Moreau envelope is the natural choice for the smoothing of both fine and coarse models.

Smoothing when $\mathrm{prox}_{g_h \circ \mathrm{D}_h}$ is known. A coarse model for the image restoration problem (5.1) is defined at iteration k of a multilevel algorithm as:

$$F_H(x_H) = (f_H \circ \mathrm{A}_H)(x_H) + ({}^{\gamma_H}(g_H \circ \mathrm{D}_H))(x_H) + \langle v_{H,k}, x_H \rangle, \tag{5.11}$$

where $v_{H,k}$ will be set to:

$$v_{H,k} = I_h^H \left[(\nabla(f_h \circ \mathrm{A}_h) + \nabla({}^{\gamma_h}(g_h \circ \mathrm{D}_h)_h)(y_{h,k})\right] - (\nabla(f_H \circ \mathrm{A}_H) + \nabla({}^{\gamma_H}(g_H \circ \mathrm{D}_H)))(x_{H,k,0}).$$

As the proximity operator is available explicitly, a non-smooth coarse model may also be used (replace ${}^{\gamma_H}(g_H \circ \mathrm{D}_H)$ by $g_H \circ \mathrm{D}_H$). We will test the two options in our experiments (Section 5.4).

Smoothing when $\text{prox}_{g_h \circ \mathrm{D}_h}$ is unknown. In the cases we are interested in, $\text{prox}_{g_h \circ \mathrm{D}_h}$ is unknown, but prox_{g_h} is available. As a result, instead of directly using the Moreau envelope of $g_h \circ \mathrm{D}_h$, we first compute the Moreau envelope of g_h and compose it with D_h. This smoothing satisfies Definition 19:

Lemma 15. *${}^{\gamma}g_h \circ \mathrm{D}_h$ is a smoothed convex function approximating $g \circ \mathrm{D}$ in the sense of Definition 19.*

Proof. Remark that ${}^{\gamma}g_h$ is a smooth convex function in the sense of Definition 19 [132]. By [132, Lemma 2.2], the fact that ${}^{\gamma}g_h \circ \mathrm{D}_h$ is a smooth function applied to a linear transformation concludes the proof. □

This smooth approximation has the following interesting property:

Lemma 16. ***[155, Lemma 3.2].*** *For any $x \in \mathbb{R}_h^N$, $\mathrm{D}_h : \mathbb{R}^{N_h} \to \mathbb{R}^{K_h}$ and $g_h : \mathbb{R}^{K_h} \to \mathbb{R}$ a convex, l.s.c., and proper function, we have that:*

$$\nabla \left({}^{\gamma}g_h \circ \mathrm{D}_h\right)(x) = \gamma^{-1}\mathrm{D}_h^* \left(\mathrm{D}_h x - \text{prox}_{\gamma g_h}(\mathrm{D}x)\right). \tag{5.12}$$

Proof. The proof is a direct consequence of the properties of the gradient operator. □

This means that an explicit form of $\text{prox}_{\gamma_h g_h}$ is sufficient to express the gradient of ${}^{\gamma_h}g_h(\mathrm{D}_h \cdot)$. A coarse model for the image restoration problem (5.1) is defined at iteration k of a multilevel algorithm as:

$$(\forall x_H \in \mathbb{R}^{N_H}), \quad F_H(x_H) = (f_H \circ \mathrm{A}_H)(x_H) + \left({}^{\gamma_H}g_H \circ \mathrm{D}_H\right)(x_H) + \langle v_{H,k}, x_H \rangle, \tag{5.13}$$

where $v_{H,k}$ will be set to:

$$v_{H,k} = I_h^H \left[\left(\nabla(f_h \circ \mathrm{A}_h) + \nabla({}^{\gamma_h}g_h \circ \mathrm{D}_h)\right)(y_{h,k})\right] - \left(\nabla(f_H \circ \mathrm{A}_H) + \nabla({}^{\gamma_H}g_H \circ \mathrm{D}_H)\right)(x_{H,k,0}).$$

5.4 Selecting the hyperparameters

In this section, we present a numerical study on the impact of some hyperparameters of the multilevel algorithm, such as the number of times the coarse level models are used, the minimization strategy at coarse level, and the number of iterations at each level. We place ourselves in some high degradation scenarios, which are the most interesting ones. The optimization problem we solve cannot be considered as part of the state-of-the-art in image restoration, as we will consider a wavelet penalized deblurring problem. Nevertheless, this problem allows to compare our inertial multilevel algorithm to FISTA in a context where all the variables are controlled, and where the inexactness cannot influence the results.

Our final choice of hyperparameters will not be guided only by the consistency of the performance of IML FISTA across all contexts. We want to bring to light a configuration of IML FISTA that is good enough for a large set of problems, instead of the best for a given problem. Such configuration should thus be more robust when applying IML FISTA to other problems.

5.4.1 Experimental setup

Dataset and degradation. We consider a large gray image (Figure 5.1 left) of size 2048×2048, yielding $N = (2^J)^2 \simeq 4\times10^6$ with $J = 11$. The linear degradation operator A_h is constructed with HNO [3] as a Kronecker product with Neumann boundary conditions, and we add a Gaussian noise (see the legend of Figure 5.1 for details). In all tests, the regularization parameter λ_h was chosen by a grid search, in order to maximize the Signal-to-Noise-Ratio (SNR) of $\widehat{x}$ obtained with FISTA at convergence. For all experiments in this manuscript, the same procedure will be used to find λ_h. Also, we initialize x_0 with the Wiener filtering of z.

Problem formulation. A usual choice for the optimization problem, even though a bit outdated now, is to apply the l_1-norm on the coefficients raised by a wavelet transform $\mathrm{D} \in \mathbb{R}^{K\times N}$, thus promoting the sparsity of the solution [20].

Given a regularization parameter $\lambda > 0$, the associated minimization problem reads:

$$\widehat{x} \in \underset{x\in\mathbb{R}^N}{\mathrm{Argmin}} \frac{1}{2}\|\mathrm{A}x - z\|_2^2 + \lambda\|\mathrm{D}x\|_1. \tag{5.14}$$

Thus, $f_h \circ \mathrm{A}_h = \frac{1}{2}\|A_h \cdot -z_h\|^2$ (with $\mathrm{A} := \mathrm{A}_h$). The penalty term $g_h \circ \mathrm{D}_h = \lambda\|\mathrm{D}_h \cdot\|_1$ is defined using a full wavelet decomposition over J levels ($\mathrm{D} := \mathrm{D}_h$).

Coarse level construction. At iteration k, the *non-smooth* coarse model F_H is defined as

$$F_H(x_H) = f_H(\mathrm{A}_H x_H) + g_H(\mathrm{D}_H x_H) + \langle v_{H,k}, x_H\rangle \tag{5.15}$$

where $\mathrm{A}_H = I_h^H \mathrm{A}_h I_H^h$ and D_H is a decomposition over $J-1$ up to $J-4$ levels, with $\lambda_H = \lambda_h/4$. Therefore, $f_H = \frac{1}{2}\|\cdot x_H - z_h\|^2$ and $g_H = \lambda_H\|\cdot x_H\|_1$.

$$\begin{aligned} v_{H,k} =& I_h^H\left(\nabla(f_h \circ \mathrm{A}_h)(y_{h,k}) + \nabla({}^{\gamma_h}(g_h \circ \mathrm{D}_h))(y_{h,k})\right) \\ &- \left(\nabla(f_H \circ \mathrm{A}_H)(x_{H,k,0}) + \nabla({}^{\gamma_H}(g_H \circ \mathrm{D}_H))(x_{H,k,0})\right). \end{aligned} \tag{5.16}$$

The third term in (5.15) enforces the first order coherence between a smoothed coarse objective function

$$F_{H,\gamma_H}(x_H) = f_H(\mathrm{A}_H x_H) + {}^{\gamma_H}(g_H \circ \mathrm{D}_H)(x_H) + \langle v_{H,k}, x_H\rangle \tag{5.17}$$

and a smoothed fine objective function F_{h,γ_h} [90] near $x_{H,k,0}$:

$$\nabla F_{H,\gamma_H}(x_{H,k,0}) = I_h^H \nabla F_{h,\gamma_h}(y_{h,k}). \tag{5.18}$$

Multilevel architecture. We use a 5-levels hierarchy: from 2048×2048 ($J = 11$) to 128×128 (indexed by $J-4$). We choose I_h^H as the low scale projection on a Symlet wavelet with 10 vanishing moments and $I_H^h = \frac{1}{4}(I_h^H)^T$. The Moreau envelope parameter associated with g_H is set to $\gamma_H = 1.1$ while γ_h is set to 1, but both values do not seem to be critical here.

Minimization operator Φ_H. At the coarse level we can decide to consider either the non-smooth approximation (5.15) of the objective function or the smoothed version (5.17). Both cases lead to a decrease in F_{H,γ_H}: indeed, taking the Moreau envelope of g_H in $F_H(x_{H,k,m}) \leq F_H(x_{H,k,0})$ yields $F_{H,\gamma_H}(x_{H,k,m}) \leq F_{H,\gamma_H}(x_{H,k,0})$. The two cases are linked by the same choice of the correction term to ensure the coherence between the two levels (5.16). We consider here three different strategies :

1. Gradient steps on the smoothed F_{H,γ_H}:
   $\Phi_S^H = (\mathrm{Id} - \tau_H(\nabla(f_H + {}^{\gamma_H}g_H) + v_H))$

2. Proximal gradient steps on the non-smooth F_H:
   $\Phi_{FB}^H = \mathrm{prox}_{\tau_H g_H}(\mathrm{Id} - \tau_H(\nabla f_H + v_H))$.

3. FISTA steps on the non-smooth F_H with the previous proximal gradient step and where the inertia follows the Nesterov's rule [70] ensuring convergence of the iterates. Noted Φ_{FISTA}^H in the following.

Performance assessment. We measure $\mathrm{Time}_{\text{IML FISTA}}$, the CPU time needed to reach $5, 2, 1, 0.1$ and 0.01% of the distance $\|F_h(x_{h,0}) - F_h(\widehat{x})\|$, where $\widehat{x}$ is computed beforehand by FISTA, and we compare it to $\mathrm{Time}_{\text{FISTA}}$, the CPU time of FISTA with the following relationship

$$\frac{\mathrm{Time}_{\text{IML FISTA}} - \mathrm{Time}_{\text{FISTA}}}{\mathrm{Time}_{\text{FISTA}}} \times 100. \tag{5.19}$$

A similar measure was used in [90] to assess the performance of MPGM with respect to other algorithms.

5.4.2 Benchmark results

We tested the performance for several values of m, and among our numerous numerical experiments, $m = 5$ at the different coarse levels appears to be a good compromise whatever the noise and blur levels. We report in Table 5.1 and Table 5.2 the relative CPU time (Equation (5.19)) for $m = 5$ at every coarse levels. In this table we evaluate two phenomena in particular: the impact of the number of coarse corrections, and the impact of the degradation level on the performance.

The impact of p. In our numerical experiments we only consider $p = 1$ (•) or $p = 2$ (•) uses of the coarse models, performed at the beginning of the iterative process. They allow to quickly determine the low frequencies components of the solution at the fine level. The choice of p depends on the sought accuracy. If a rough approximation is sufficient, fixing $p = 1$ is the best choice, while $p = 2$ is better for lower thresholds. While we obtain good gains for those, for very low ones the use of a multilevel strategy is not useful, but note that it does not deteriorate the performance either.

The impact of noise and blur level. For all three minimization methods at coarse level acceleration increases significantly as the blur gets worse. Moreover, as the noise decreases, the improvement obtained with $\Phi_{H,FISTA}$ as compared to others Φ_H increases.

Noise \ Blur		(a) size(blur) = [40, 40], σ(blur) = 7.3				
(1) $\sigma = 0.01$	FISTA CPU time	16	28	42	161	401
	$\Phi_{H,S}$	-20 •	-22 •	$+1$ •	$+1$ •	-1 •
	$\Phi_{H,FB}$	-19 •	-19 •	$+5$ •	$\mathbf{+2}$ •	$\mathbf{+1}$ •
	$\Phi_{H,FISTA}$	$\mathbf{-51}$ •	$\mathbf{-32}$ •	$\mathbf{-4}$ •	$+2$ •	$+1$ •
(2) $\sigma = 0.04$	FISTA CPU time	14	22	34	108	220
	$\Phi_{H,S}$	$\mathbf{-22}$ •	-10 •	-1 •	$\mathbf{-1}$ •	-1 •
	$\Phi_{H,FB}$	-22 •	-10 •	-1 •	$+1$ •	-1 •
	$\Phi_{H,FISTA}$	-21 •	$\mathbf{-12}$ •	$\mathbf{-10}$ •	-1 •	$\mathbf{-2}$ •

Table 5.1: The first line of each subtable represents the computation time (in sec) needed by FISTA to reach $5, 2, 1, 0.1$ and 0.01% of the distance $\|F_h(x_{h,0}) - F_h(\widehat{x})\|$. Then for each type of minimization algorithm at coarse level, we display the CPU time relative to FISTA (5.19) (in %) for the best configuration with a colored bullet : $p = 1$ • and $p = 2$ •. In all cases : $m = 5$. SNR of z : (1) 11.05 (2) 9.64. SNR of $x_{h,300}$ computed by IML FISTA : (1) 12.71 (2) 11.02.

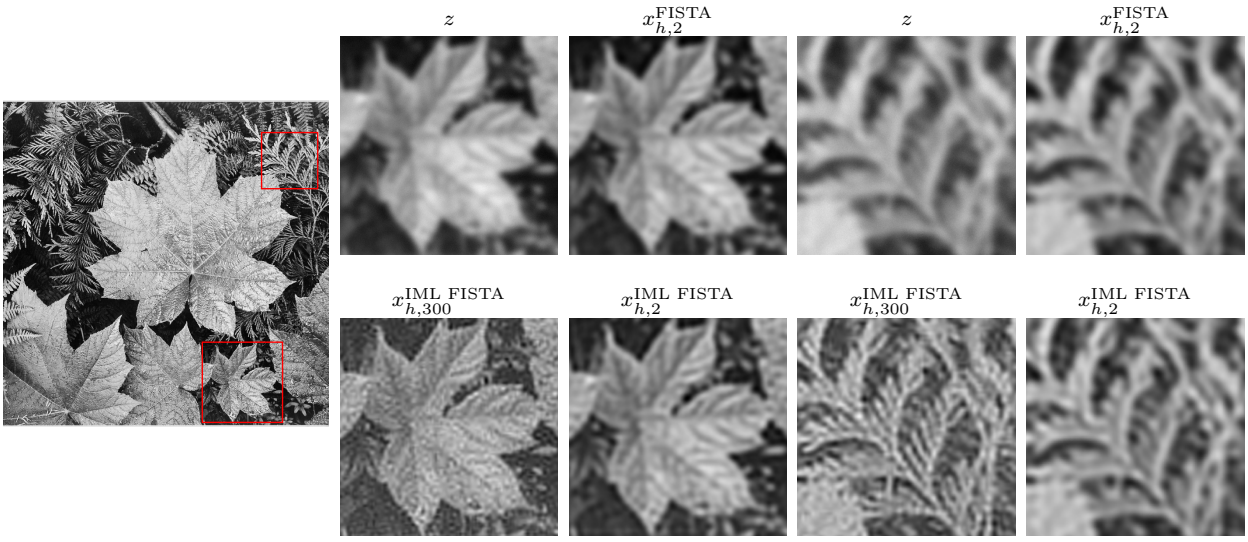

Figure 5.1: *Top* : From left to right : Original 2048×2048 image[2] $\overline{x}$, (first row) zoom of the degraded image z for a noise with $\sigma = 0.01$ and a Gaussian blur of size 40×40 and 7.3 standard deviation and of x_2^h computed by FISTA. (Second row) Zoom of x_2^h and x_{300}^h computed by IML FISTA. $\lambda_h = 1.7 \times 10^{-4}$.

Noise \ Blur		(b) size(blur) = [88, 88], σ(blur) = 16				
(1) $\sigma = 0.01$	FISTA CPU time	17	30	42	148	421
	$\Phi_{H,S}$	**−51** •	**−44** •	−18 •	**+4** •	**−1** •
	$\Phi_{H,FB}$	−50 •	−42 •	−15 •	+6 •	+1 •
	$\Phi_{H,FISTA}$	−50 •	−42 •	**−35** •	+8 •	+1 •
(2) $\sigma = 0.04$	FISTA CPU time	15	25	34	122	315
	$\Phi_{H,S}$	−29 •	−25 •	−18 •	**+3** •	**+1** •
	$\Phi_{H,FB}$	−42 •	−31 •	−16 •	+5 •	+2 •
	$\Phi_{H,FISTA}$	**−42** •	**−31** •	**−22** •	+7 •	+2 •

Table 5.2: The first line of each subtable represents the computation time (in sec) needed by FISTA to reach $5, 2, 1, 0.1$ and 0.01% of the distance $\|F_h(x_{h,0}) - F_h(\widehat{x})\|$. Then for each type of minimization algorithm at coarse level, we display the CPU time relative to FISTA (5.19) (in %) for the best configuration with a colored bullet : $p = 1$ • and $p = 2$ •. In all cases : $m = 5$. SNR of z : (1) 11.03 (2) 9.63. SNR of $x_{h,300}$ computed by IML FISTA : (1) 12 (2) 10.6.

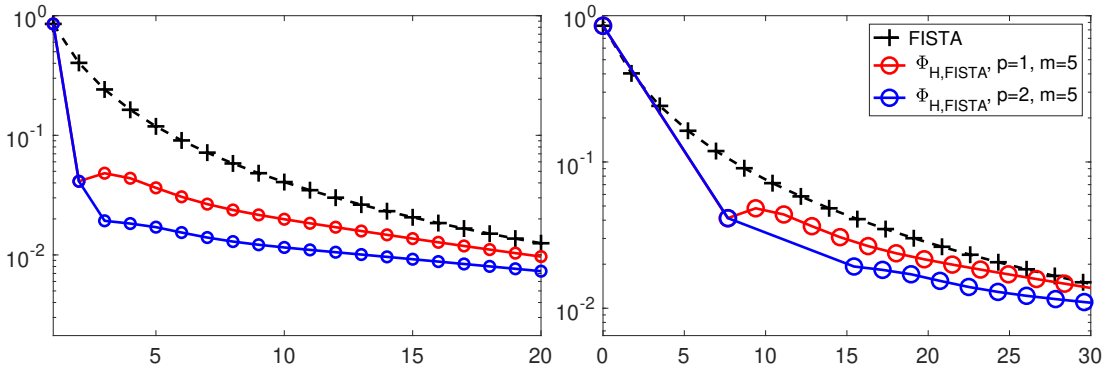

Figure 5.2: (Left) Evolution of F_h versus iterations for IML FISTA with Φ^H_{FISTA} for $p = 1, 2$, $m = 5$. (Right) Same for CPU time (in sec). Degradation: Gaussian noise with $\sigma_{\text{noise}} = 0.01$ and a Gaussian blur of size 40×40 and 7.3 standard deviation. $\lambda_h = 1.7 \times 10^{-4}$.

The main takeaway from these experiments is that, with a few coarse corrections, our method can provide good approximations of the solution while staying competitive with FISTA for high precision approximations. Guided by these results, we will choose $m = 5$ and $p = 2$ for the rest of our experiments in this chapter. Moreover, there seems to be no significant difference between smooth and non-smooth coarse models, so we will use the smooth version in the following for simplicity. Using extrapolation steps at coarse level, even though it seems to yield the most gains in terms of CPU time, won't be used anymore, again for simplicity. It is possible that all of our results in the following may be improved with these steps, but we are primarily interested in constructing a robust algorithm, that performs well on any given problem, rather than finding the best configuration for each specific problem.

Finding the best configuration is obviously interesting, but our goal is to propose an algorithm that generalizes well to other problems, and thus the best configuration for one problem may be completely different for another. In practice no one would tune its algorithm for hours to gain a few % of CPU time on one run, so we will not do it either.

Starting from these choices of hyperparameters, we will now test the robustness of our algorithm on color image restoration problems, where state-of-the-art regularizations are used.

5.5 Application to color image restoration

In this section, based on the hyperparameters identified in the previous section, we conduct extensive numerical experiments to validate the design of IML FISTA. We consider two types of image restoration problems: deblurring and inpainting (both with additive Gaussian noise). Our goal will be twofold: test on high dimensional problems, with state-of-the-art regularization (TV and NLTV), to see how our algorithm behaves if we use the same parameters on all problems; and test on problems of smaller dimension, in the same setting, to see if our algorithm is competitive. For the latter, we expect IML FISTA to perform less and less well as the dimension decreases, but this in fact not true for all degradation levels.

These experiments will also consolidate our vision about the tuning of IML FISTA's hyperparameters: choosing the optimal ones is not necessary, our algorithm is robust enough to provide good results in a wide range of contexts.

5.5.1 Experimental setup

Degradation types. We consider two types of image reconstruction problems: a restoration problem where the linear operator A is a Gaussian blur, and an inpainting problem where A models the action of random pixel deletion. In all cases, we consider an additive white Gaussian noise with standard deviation σ.

Minimization problem. At fine level, we consider the state-of-the-art optimization problem in this context, the minimization of the sum of a quadratic data-fidelity term and a sparsity prior based on a total variation $\ell_{1,2}$-norm (isotropic total variation):

$$(\forall x \in \mathbb{R}^{N_h}), \qquad F_h(x) = \frac{1}{2}\|\mathrm{A}_h x - z_h\|_2^2 + \lambda_h \|\mathrm{D}_h x\|_{1,2}, \tag{5.20}$$

with $\lambda_h > 0$. In all the experiments, the regularization parameter λ_h was chosen by a grid search, in order to maximize the SNR of $\widehat{x}$ computed by FISTA at convergence. Finally, we choose as initialization $x_{h,0}$, the Wiener filtering of z.

Estimation of the proximity operator. The minimization of the dual problem associated with the proximity operator of $x \mapsto \|\mathrm{D}_h x\|_{1,2}$ is carried out by FISTA coupled with a warm start strategy as in [57]. We set the initialization value of *tol* (in Algorithm 3) based on the reconstruction quality of images in a Total Variation based denoising problem (that is equivalent to one computation of the associated proximity operator). $tol = 10^{-8}$ at the start of the optimization unless stated otherwise.


Figure 5.3: ImageNet Car "ILSVRC2012_test_00000164"[1]. Pillars of Creation[2]. Credits: SCIENCE: NASA, ESA, CSA, STScI (Image processing): Joseph DePasquale (STScI), Alyssa Pagan (STScI), Anton M. Koekemoer (STScI).

Experiment datasets. We consider two color images of different sizes to evaluate the impact of the problem's dimension: "ImageNet Car" the picture of a yellow car of size $512 \times 512 \times 3$, taken from the ImageNet dataset, and a picture taken by the James Webb Space Telescope with its Near-Infrared Camera and its Mid-Infrared Instrument of the structure called "Pillars of Creation" of size $2048 \times 2048 \times 3$ (Figure 5.3). Pixels values are normalized so that the maximum value across all channels is 1.

Multilevel structure. For all our experiments we use a 5-levels hierarchy. For the image "Pillars of Creation", the first level corresponds to an image of size $2048 \times 2048 \times 3$, and the fifth level to an image of size $128 \times 128 \times 3$. Similarly, for "ImageNet Car" the first level corresponds to an image of size $512 \times 512 \times 3$ and the fifth level to an image of size $32 \times 32 \times 3$.

The coarse model associated to (5.20) is written as:

$$(\forall x \in \mathbb{R}^{N_H}), \qquad F_H(x) = \frac{1}{2}\|\mathrm{A}_H x - z_H\|_2^2 + \lambda_H \left({}^{\gamma_H} g_H \left(\mathrm{D}_H x\right)\right) + \langle v_H, x \rangle, \tag{5.21}$$

with $\lambda_H > 0$, $z_H = I_h^H z_h$ and g_H the $\ell_{1,2}$-norm applied on the N_H components of $\mathrm{D}_H x$, as for the fine level.

As the dimension of the problem is reduced by a factor 4 every time we lower the resolution, we set the regularization parameter λ_H at coarse level to a quarter of the value of the regularization parameter at the next higher level. In practice, this ratio gives the best performance in terms of decrease of the fine level objective function. The CIT operators were built for every pair of levels with "Symlet 10" wavelets corresponding to a filter size of 20 coefficients.

Remark 12. *Ideally, in order to speed up the computations, one would like to choose an approximation R_H whose proximity operator is known under closed form, even when R_h does not possess this desirable property. However, we have seen in our experiments that choosing R_H not faithful to R_h deteriorates the performance of the multilevel algorithm. For instance, when R_h is the TV based norm, choosing a Haar wavelet based norm for R_H is suboptimal, even though there is a link between Haar wavelet and total variation thresholdings [30, 31].*

Finally, recall that our algorithm take the compressed form:

$$\begin{aligned} \bar{y}_{h,k} &= \mathbf{ML}(y_{h,k}), \\ x_{h,k+1} &= \mathrm{FB}_i^{\epsilon_{h,k}}(\bar{y}_{h,k}), \\ y_{h,k+1} &= x_{h,k+1} + \alpha_{h,k}(x_{h,k+1} - x_{h,k}) \end{aligned} \tag{5.22}$$

where

- $0 < \tau_{h,k} < 1/\beta_h$
- $\alpha_{h,k} = \frac{t_k - 1}{t_{k+1}}$
- $t_k = \left(\frac{k+a-1}{a}\right)^d$, with $d \in (0, 1]$ and $a > \max\{1, (2d)^{\frac{1}{d}}\}$ [70, Definition 3.1].

5.5.2 Application to image deblurring

In this section, we consider the problem of image deblurring with additive Gaussian noise. We compare the performance of IML FISTA with the one of FISTA. First, we confirm that FISTA outperforms FB even in this inexact proximal context. Then, we show that IML FISTA outperforms FISTA in terms of convergence speed and quality of the reconstruction.

Experimental setup. To get a full picture of the performance of IML FISTA, we propose four scenarios, corresponding to four different combinations of the size of the Gaussian blur PSF and of the value of the standard deviation σ(noise) of the Gaussian noise. These four scenarios are described in Table 5.3.

Blur \ Noise	$\sigma(\text{noise}) = 0.01$	$\sigma(\text{noise}) = 0.05$
$\dim(\text{PSF}) = 20$, $\sigma(\text{PSF}) = 3.6$	low blur, low noise	low blur, high noise
$\dim(\text{PSF}) = 40$, $\sigma(\text{PSF}) = 7.3$	high blur, low noise	high blur, high noise

Table 5.3: Four scenarios of Gaussian blur degradation with additive Gaussian noise.

FB/FISTA vs IML FB/FISTA. This first set of experiments allows us to compare several formulations of IML FISTA, including its particular instances FB and FISTA. Algorithm 5.22 can take the form of

- FB when $d = 0$ and $\mathbf{ML}(y_{h,k}) = y_{h,k}$,
- IML FB when $d = 0$ and $\mathbf{ML}(y_{h,k}) = \bar{y}_{h,k}$,
- FISTA when $d = 1$ and $\mathbf{ML}(y_{h,k}) = y_{h,k}$,
- IML FISTA when $d = 1$ and $\mathbf{ML}(y_{h,k}) = \bar{y}_{h,k}$.

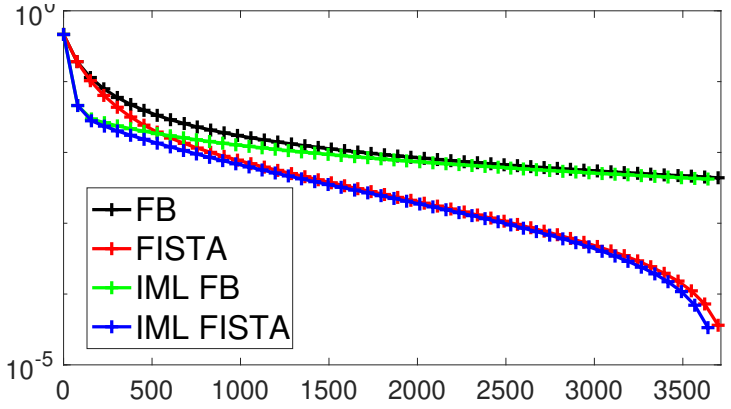

Figure 5.4: Comparison of FB and FISTA against their multilevel counterpart constructed with our framework, IML FB and IML FISTA for the restoration $\ell_{1,2}$-TV problem for the Pillars of Creation image (see top left corner Table 5.3). To put the emphasis on the performance's difference between these algorithms, the objective function evolution is displayed in a log scale between the initial value and the minimum value obtained by these four algorithms in 50 iterations.

In Figure 5.4, we focus on the top left corner degradation configuration (Table 5.3) and display the evolution of the objective function w.r.t. the CPU time for the four versions of Algorithm 5.22. We observe that IML FB (resp. FISTA) converges faster than FB (resp. FISTA) and additionally, it confirms that FISTA and IML FISTA outperform forward-backward approaches without inertial steps. In the following experiments, we focus on FISTA and IML FISTA comparisons.

Experimental performance for different degradation levels. In each of the following figures, the organization of the four plots coincides with the configurations in table 5.3. For each of them, we tested two regularizations: $\ell_{1,2}$-TV and $\ell_{1,2}$-NLTV. Because the relative behavior of IML FISTA with respect to FISTA is similar for the two regularizations, for the sake of conciseness, we only report here the results obtained with the $\ell_{1,2}$-TV prior. Figure 5.5 and Figure 5.6 provide a first set of results for the 2048×2048 Pillars of Creation image. We focus in the following on the 25 first iterations as the main gain provided by the proposed method is obtained at the start of the optimization. We can see that in all cases, the decreasing of the objective function of IML FISTA is faster than that of FISTA. Given the cost of estimating proximity operators for TV and NLTV based regularizations, the computational overhead of a multilevel step is almost negligible, as we expected (cf. Figure 5.5). Thus, the two low cost coarse corrections are sufficient for our algorithm to gain an advantage that FISTA cannot recover without decreasing the tolerance on the approximation of the proximity operator. As a result, this would entail higher computation time at each iteration as the error must decrease with the number of iterations to converge. Most interestingly, if we compare the methods at the very early stages of the optimization process, after the same number of iterations, IML FISTA reaches a much lower value for the objective function, leading to a much better reconstruction. The difference is particularly striking after 2 iterations (Figure 5.6).

One can also notice that increasing the noise (and thus increasing the value of regularization term λ) degrades the relative performance of our algorithm compared to FISTA.

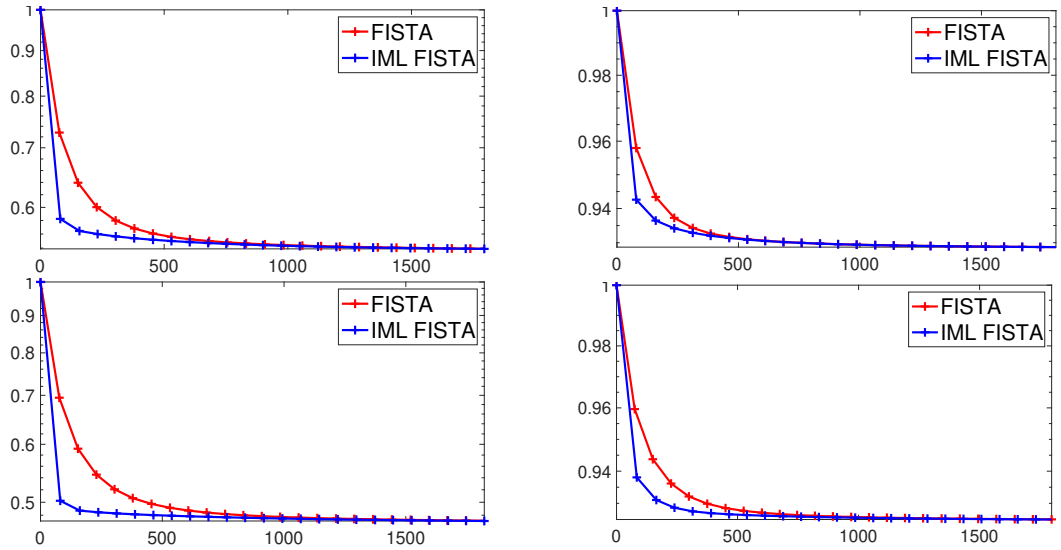

Figure 5.5: Deblurring $\ell_{1,2}$-TV for the Pillars of Creation image. Objective function (normalized w.r.t. the initial value) vs CPU time (sec). First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: $\dim(\text{PSF}) = 20$, $\sigma(\text{PSF}) = 3.6$; second row: $\dim(\text{PSF}) = 40$, $\sigma(\text{PSF}) = 7.3$. For each plot, the crosses represent iterations of the algorithm.

This behavior was observed in the exact proximal case (see previous section) albeit it is far less pronounced here. Similarly, increasing the blur size improves the relative performances of IML FISTA, just like in the case of exact expression for the proximity operator.

We stress that the potential of multilevel strategies, especially for high levels of degradation (i.e., blurring and noise), is particularly evident for large scale images: on smaller problems the overhead introduced by the method may overcome the gain obtained in passing to lower resolutions. This is evident when looking at the results obtained in the same degradation context for the Yellow Car image of size $512 \times 512 \times 3$. We reproduce in Figure 5.7 the evolution of the objective function when the regularization is the $\ell_{1,2}$-TV norm. With this problem of small dimension, the relative performances of IML FISTA compared to those of FISTA are less impressive than in the case of the Pillars of Creation image. The visual gains are also less obvious (Figure 5.8). We can still observe that for this small problem, the degradation impacts greatly the relative performances: as the blur size increases, the relative performances of IML FISTA compared to FISTA improve (bottom left curve of Figure 5.7). Thus, one can expect to have good performances with IML FISTA for small problems if the degradation is high.

5.5.3 Application to image inpainting

Here again, we consider four scenarios based on two variables: the percentage of missing pixels and the standard deviation of the Gaussian noise $\sigma(\text{noise})$. These four scenarios are specified in Table 5.4. For each of these four scenarios we tested two regularizations: $\ell_{1,2}$-TV and $\ell_{1,2}$-NLTV. In this case we only report the results obtained with the $\ell_{1,2}$-NLTV prior.

Again, in all cases, the objective function decreases faster with IML FISTA than with

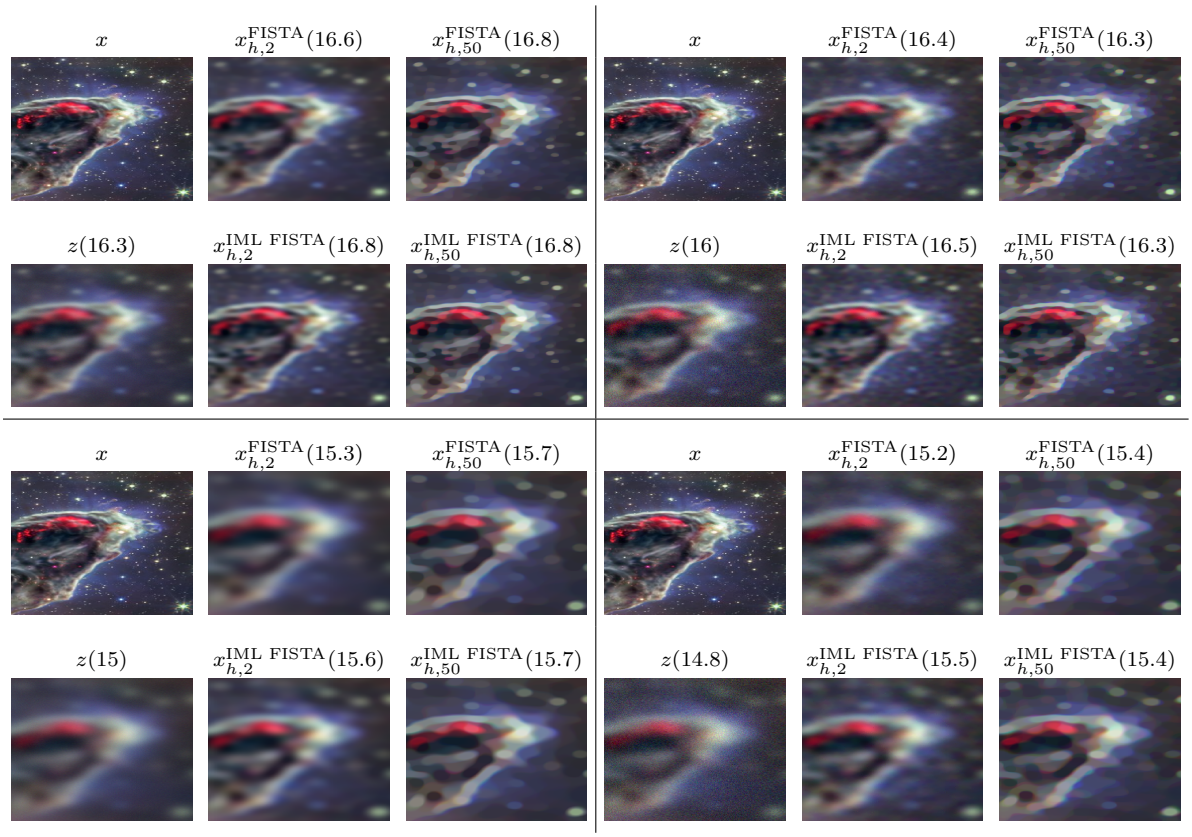

Figure 5.6: Deblurring $\ell_{1,2}$-TV for the Pillars of Creation image. Small crop of the image after 2 iterations and after 50 iterations for FISTA (top row) and IML FISTA (bottom row) compared to the original (x) and degraded (z) images. For each image we report the SNR in dB. First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: $\dim(\text{PSF}) = 20$, $\sigma(\text{PSF}) = 3.6$; second row: $\dim(\text{PSF}) = 40$, $\sigma(\text{PSF}) = 7.3$.

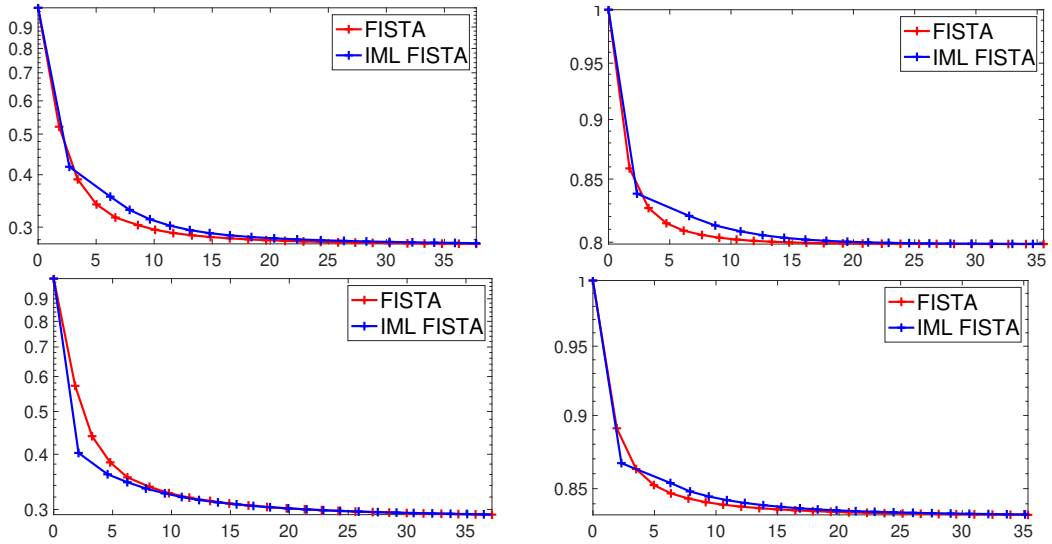

Figure 5.7: Deblurring $\ell_{1,2}$-TV for the Yellow Car image (small dimensional image). Objective function (normalized with initialization value) vs CPU time (sec). First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: $\dim(\text{PSF}) = 20$, $\sigma(\text{PSF}) = 3.6$; second row: $\dim(\text{PSF}) = 40$, $\sigma(\text{PSF}) = 7.3$. For each plot, the crosses represent iterations of the algorithm.

Inpainting \ Noise	$\sigma(\text{noise}) = 0.01$	$\sigma(\text{noise}) = 0.05$
missing pixels 50%	low inpainting, low noise	low inpainting, high noise
missing pixels 90%	high inpainting, low noise	high inpainting, high noise

Table 5.4: Four scenarios of inpainting degradation with additive Gaussian noise.

FISTA, proving that the computational cost of multilevel steps is almost negligible. The two performed coarse corrections bring a considerable advantage to the minimization achieved with IML FISTA (Figure 5.9). Also, one can remark that given a capped sub-iterations budget, IML FISTA reaches the smallest possible value, faster than FISTA. Comparing the two methods after only two iterations, is particularly convincing as we can observe it in Figure 5.10: IML FISTA has already recovered the main features of the original image, while FISTA is still far from it. Moreover, as it was already the case for the deblurring task, IML FISTA outperforms FISTA in terms of convergence speed, specifically when the original image is heavily corrupted. As for the deblurring task, we display the results under the same degradation contexts (i.e., inpainting and noise) for the Yellow Car image. We reproduce in Figure 5.11 the evolution of the objective function when the regularization is the $\ell_{1,2}$-NLTV norm and in Figure 5.12 the reconstructed images. In contrast to the deblurring case, IML FISTA still performs better than FISTA for an inpainting task on a relatively small image size. This suggests that the dependency of IML FISTA's performances to the problem dimension is clearly linked to the degradation context.

Figure 5.8: Deblurring $\ell_{1,2}$-TV for the Yellow Car image. Small crop of the image after 2 iterations and after 50 iterations for FISTA (top row) and IML FISTA (bottom row) compared to the original (x) and degraded (z) images. For each image we report the SNR in dB.
First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: $\dim(\text{PSF}) = 20$, $\sigma(\text{PSF}) = 3.6$; second row: $\dim(\text{PSF}) = 40$, $\sigma(\text{PSF}) = 7.3$.

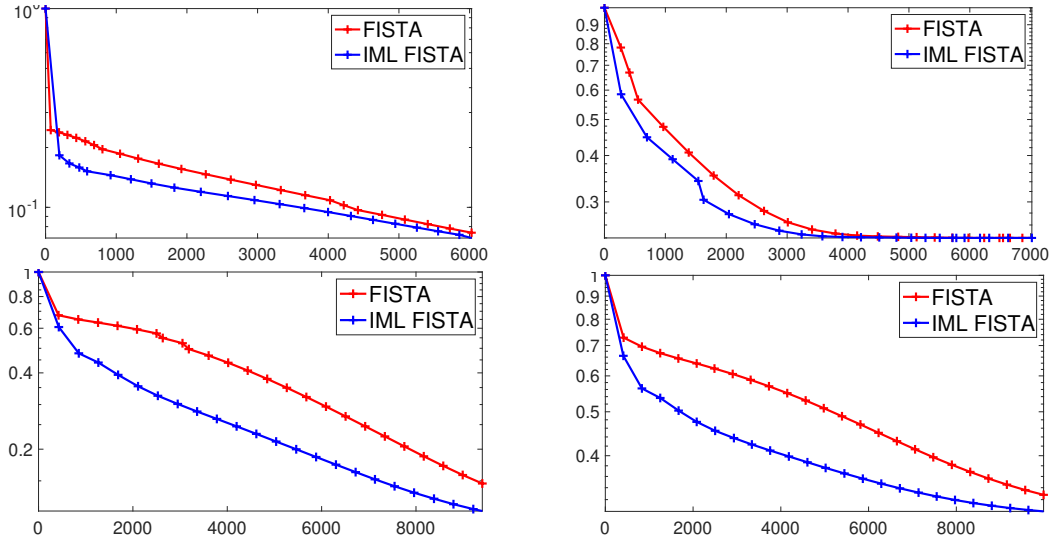

Figure 5.9: Inpainting $\ell_{1,2}$-NLTV for the Pillars of Creation image. Objective function (normalized with initialization value) vs CPU time (sec). First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: missing pixels 50%; second row: missing pixels 90%. For each plot, the crosses represent iterations of the algorithm.

Discussion of the results. With these two tasks, we have highlighted the robustness of our algorithm to the degradation context, in the presence of inexactness on the proximity operator. IML FISTA has the potential to outperform FISTA in a wide range of context, and can even be interesting for low dimensional problems. With that said, the main aim of multilevel optimization is to tackle high dimensional problems. An easy way to increase the dimension of the problem is to consider hyperspectral images, which we will address in the next section.

5.6 Application to hyperspectral image restoration

We conclude this experimental chapter by applying IML FISTA to a hyperspectral image (HSI) restoration problem.

5.6.1 Experimental setup

The acquisition of hyperspectral images is of tremendous importance in many fields such as remote sensing [156] or art analysis [157,158]. It is often impaired by missing data and noise due to cameras defect, and blurring effects. Several methods have been designed to handle them [159]. Among them, the variational approach is of great interest but suffers from a high computational cost [159]. This approach is a particular case of Problem (5.1) where

$$F_h(x) = \frac{1}{2}\|\mathrm{A}_h x - z\|_2^2 + \lambda \sum_{i=1}^{N_h} \|(\mathrm{D}_h x)^i\|_*, \tag{5.23}$$

where $\|\cdot\|_*$ is the nuclear norm. The proximity operator of this norm is a soft-thresholding operation on the singular values of $\mathrm{D}_h x$ [33]. The nuclear norm allows us to take into account the strong correlation between the bands to improve the reconstruction.

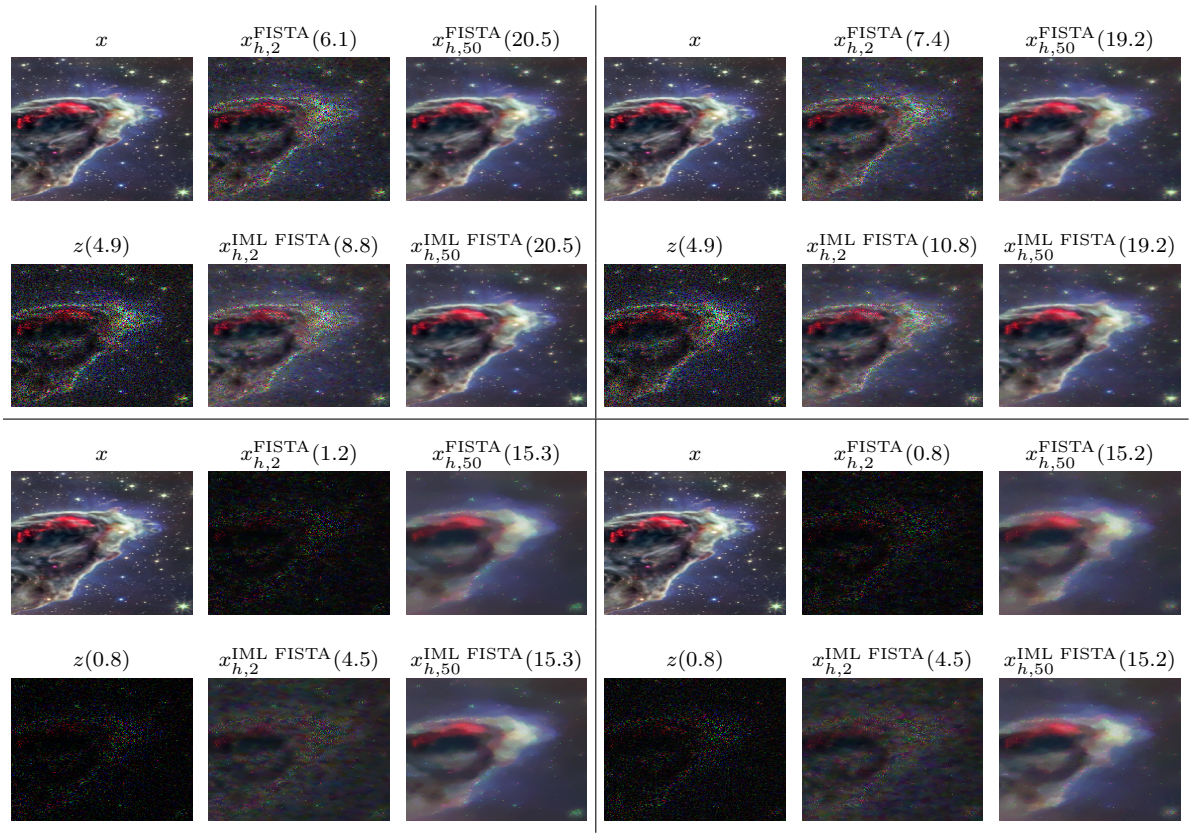

Figure 5.10: Inpainting $\ell_{1,2}$-NLTV for the Pillars of Creation image. Small crop of the image at 2 iterations and after 50 iterations for FISTA (top row) and IML FISTA (bottom row) compared to the original (x) and degraded (z) images. For each image we report the SNR in dB. First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: missing pixels 50%; second row: missing pixels 90%.

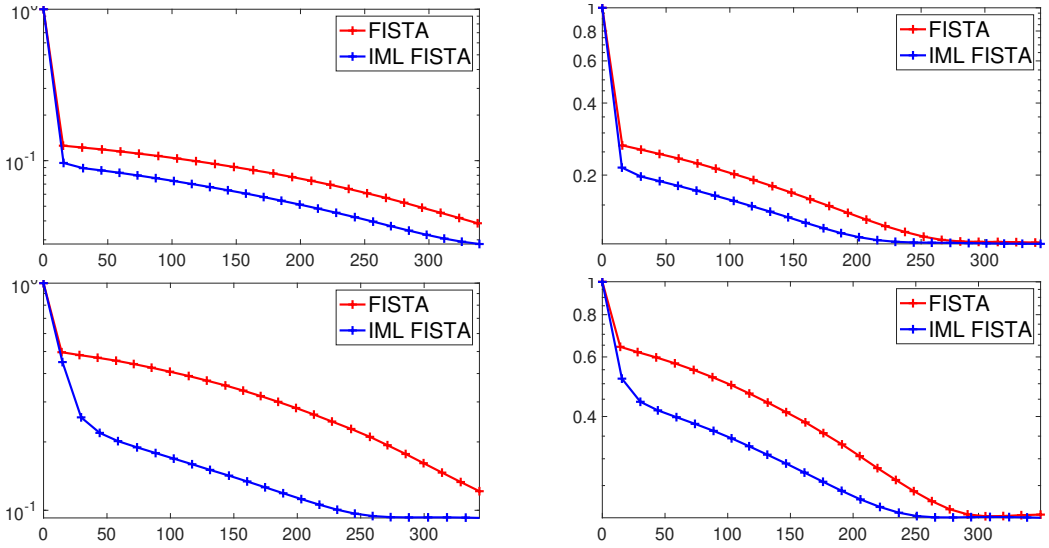

Figure 5.11: Inpainting $\ell_{1,2}$-NLTV for the Yellow Car image. Objective function (normalized with initialization value) vs CPU time (sec). First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: missing pixels 50%; second row: missing pixels 90%. For each plot, the crosses represent iterations of the algorithm.

Here a coarse level can be derived naively from the nature of those images: high correlation between bands is observed on hyperspectral images, and thus it seems natural to exploit this redundancy to reduce the dimension and restore the image.

Notations. Formally, we denote $x^{(i,b)} = x^{(i_1,i_2,b)}$ the pixel located at the spatial index $i = (i_1, i_2) \in \{1, \dots, N_r\} \times \{1, \dots, N_c\}$ and band $b \in \{1, \dots, L\}$ of HSI x. x can be represented as a hypercube of size $N_h = L \times N_r \times N_c$. We denote $w^{(b)}$ the wavelength associated with the b band. We also note $\mu(\lambda)$ the mean of the differences $\lambda^{(b+1)} - \lambda^{(b)}$ for all b and $\sigma(\lambda)$ the associated standard deviation.

5.6.2 Information transfer operators

Hyperspectral images naturally present a redundancy of spatial and/or spectral information [160–162]. We propose to exploit these two correlations to define coarse approximations of Problem (5.23).

Dimension reduction along the spatial dimension. We aim to reduce the size of an HSI by reducing the size of each band, with the same procedure we use for color images. Formally,

$$(\forall b = \{1, \dots, L_h\}) \quad x_H^{(:,b)} = I_h^H(x_h^{(:,b)}) \tag{5.24}$$

where I_h^H is defined as in Equation (5.9), with a Symlet 10 wavelet filter.

Dimension reduction along the spectral dimension. We aim to reduce the size of an HSI by reducing the number of bands. A small wavelength difference between two successive bands suggests a strong correlation between them. This similarity can be

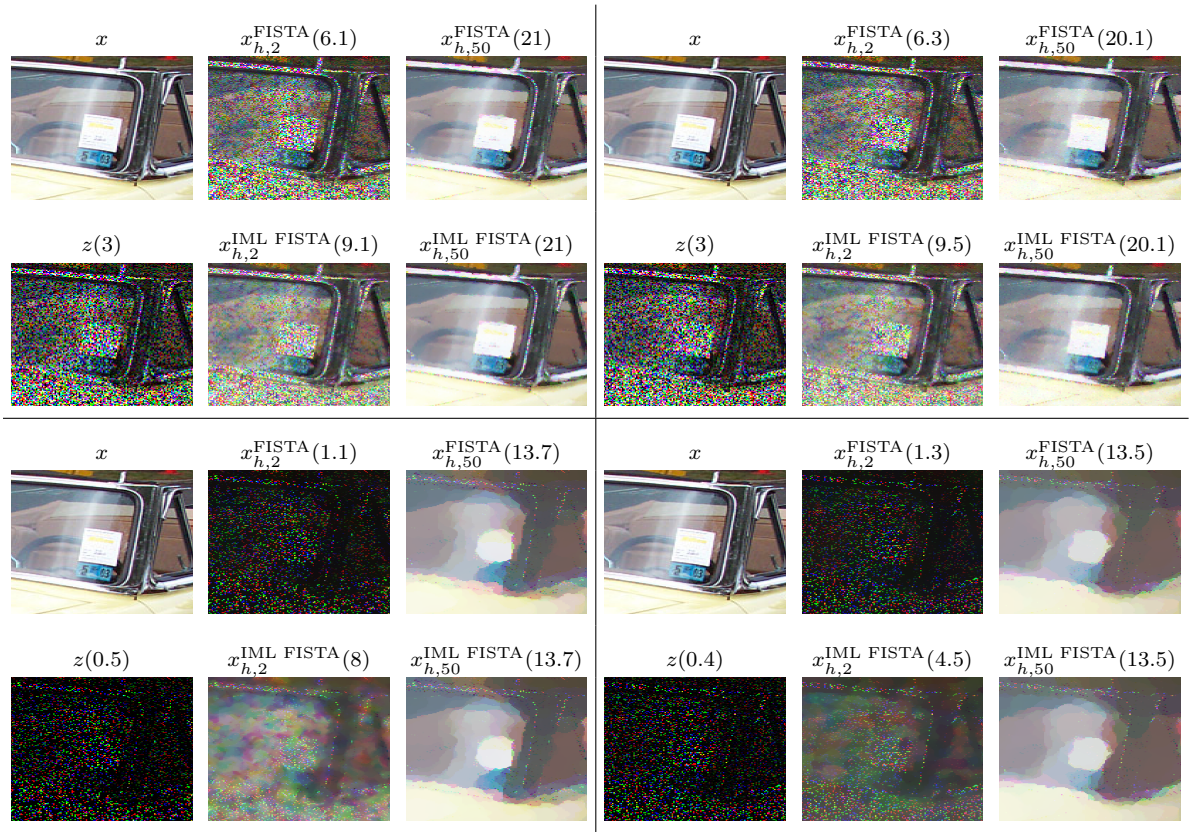

Figure 5.12: Inpainting $\ell_{1,2}$-NLTV for the Pillars of Creation image. Small crop of the image at 2 iterations and after 50 iterations for FISTA (top row) and IML FISTA (bottom row) compared to the original (x) and degraded (z) images. For each image we report the SNR in dB. First column: $\sigma(\text{noise}) = 0.01$; second column: $\sigma(\text{noise}) = 0.05$. First row: missing pixels 50%; second row: missing pixels 90%.

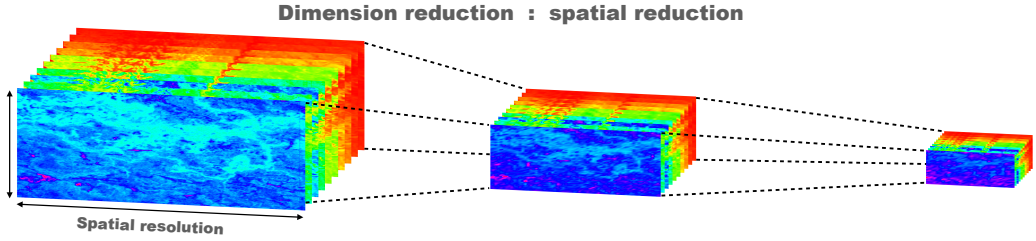

Figure 5.13: Illustration of the dimension reduction process for an HSI of size $256 \times 256 \times 145$ when reducing the dimension of each band (hence the name spatial reduction) without touching the number bands. For illustration purposes, each band is not represented here.

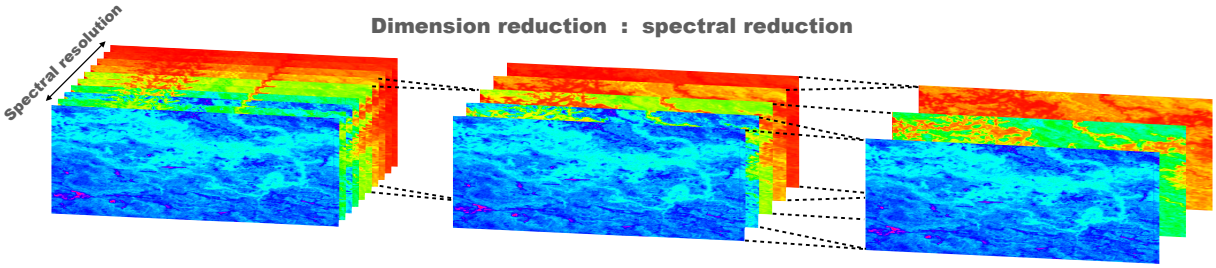

Figure 5.14: Illustration of the dimension reduction process for an HSI of size $256 \times 256 \times 145$ when reducing the dimension along the bands (hence the name spectral reduction) without touching the size of each band. For illustration purposes, each band is not represented here.

difficult to measure in our case (for a review of methods see [161]) because the observed HSI is very degraded. We have therefore chosen a simple heuristic to infer this correlation, independent of the content of the band. For all $b \in \{1, \dots, L\}$, every two consecutive bands whose wavelengths difference is smaller than $\mu(\Delta) + \sigma(\Delta)$ then these two bands are a priori correlated and will be aggregated at the coarse level. The "distant" bands in wavelength are kept as they are at the coarse level. There will therefore be L_H bands at the coarse level, of size $N_{h,r} \times N_{h,c}$, with $L_H \leq L_h$. Algorithm 8 details this process (see also Figure 5.14).

Algorithm 8 Band aggregation

$b, \ell = 1$
while $b \leq L_h - 1$ **do**
  if $\lambda^{(b+1)} - \lambda^{(b)} \leq \mu(\lambda) + \sigma(\lambda)$ **then**
    $x_H^{(:,\ell)} = \frac{1}{2}(x_h^{(:,b)} + x_h^{(:,b+1)})$,
    $\ell = \ell + 1$
  else
    $x_H^{(:,\ell)} = x_h^{(:,b)}$,
    $x_H^{(:,\ell+1)} = x_h^{(:,b+1)}$,
    $\ell = \ell + 2$
  end if
  $b = b + 2$
end while

We apply the same operation on A_h by averaging the blocks that represent the bands.

5.6.3 Application to inpainting

Here we assess the relative performances of the two strategies designed to reduce the dimension of the hyperspectral images.

Degradation. The operator A will model the missing pixels: 50% of the pixels in each band are randomly set to 0. Therefore, each band is degraded independently of the others. Gaussian noise of variance $\sigma = 0.01$ is then added to each band. A_h is therefore a matrix of L_h diagonal blocks where each block is a mask associated with the pixels in a band.

Dataset. The numerical experiments are carried out on three HSIs: one on the Washington DC Mall[1], one on the Okavango Delta in Botswana[2] (which is used to illustrate the information transfer operators in Figures 5.13 and 5.14) and one of a wood engraving of St Christopher[3].

Algorithm parameters. For all our experiments, we use a 5-level hierarchy (see below). We always impose $p = 2$ coarse corrections at the start of the optimization, each with $m = 5$ minimization iterations per level: our standard configuration now (Section 5.4).

When playing around with the parameters, we observed that IML FISTA was fast on this type of problems, and so the error computed on the proximity operator could not decrease fast enough with our procedure (see Algorithm 3) and induced increase of the objective functions after several iterations. To take this into account, we reduce the inertia of IML FISTA by fixing d to 0.5 (for FISTA $d = 1$). The two algorithm were stopped after a given computation time accounting for 50 iterations of FISTA, and 41 of IML FISTA.

Specifics of information transfer operators. Our information transfer operators being data dependent, we present the resulting hierarchy of approximations for each HSI.

Spatial reduction. For the engraving of St Christopher, the first level corresponds to an HSI of size $(512)^2 \times 33$, and the fifth level to an HSI of size $(32)^2 \times 33$. The reduction factor is the same for the other HSIs, giving an overall reduction factor of 256. The A_H operator is constructed by taking every other column and every other row of A_h.

Spectral reduction. For the Washington DC Mall, the first level corresponds to an HSI of size $(256)^2 \times 191$, and the fifth level corresponds to an HSI of size $(256)^2 \times 23$. For the Okavango Delta HSI, the first level corresponds to an HSI of size $(256)^2 \times 145$, and the fifth level to an HSI of size $(256)^2 \times 12$. For the HSI of St Christopher, the first level corresponds to an HSI of size $(512)^2 \times 33$, and the fifth level to an HSI of size $(512)^2 \times 3$. The A_H operator is constructed using Algorithm 8's procedure: the A_h blocks are merged or retained to define A_H.

For both information transfer methods, the D_H operator is a reduced order version of D_h: $\mathrm{D}_H \in \mathbb{R}^{\widetilde{K}L_H N_H \times N_H}$.

Impact of information transfer. As the dimension reduction is smaller for spectral information transfer, the cost of a coarse correction is higher. We also expected that it would slow down this version of IML FISTA, that we called IML FISTA Spec (for spectral), with respect to IML FISTA Spat (for spatial). These corrections will, however, bring a greater decrease in the objective function and improve the rest of the optimization.

[1] Washington DC Mall: acquisition by HYDICE [163].
[2] Okavango Delta : acquisition by NASA EO-1 satellite [164].
[3] St Christopher: acquisition by authors of [165].

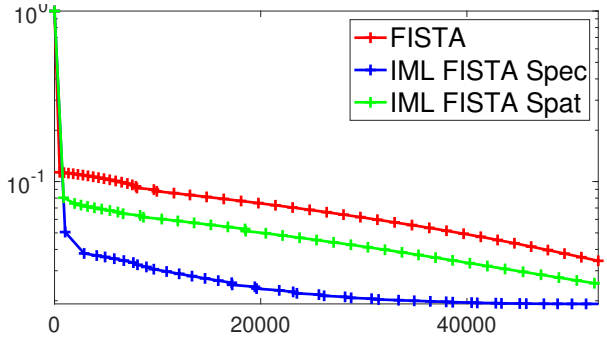

Figure 5.15: Evolution of the objective function for the inpainting problem on the St Christopher HSI with respect to the CPU time (in seconds). The green and blue curves represent the evolution of the objective function for IML FISTA Spec and IML FISTA Spat in comparison to FISTA in red.

Performance with respect to FISTA. In both cases IML FISTA is much better than FISTA at *equivalent computation time*: we can observe gains of several decibels on the SNR of the reconstructions (see Table 5.5). Figure 5.15 shows the evolution curves of the objective functions and a comparison of a channel from the St Christopher HSI. Similar curves were obtained for the other HSIs tested, we do not reproduce them here. IML FISTA Spec enables good quality restorations to be achieved on inpainting problems

	z	FISTA	IML Spat	IML Spec
Washington DC	4.8	15.8	18.6	**19.5**
Okavango Delta	5.4	18.7	23.0	**24.8**
St Christopher	5.4	21.7	32.7	**35.5**

Table 5.5: Restoration results at equal computation time (around 50 iterations) and around convergence for the three HSI. Metric: SNR (dB). In **bold**, we highlight the best result. Note that IML FISTA Spat also vastly outperforms FISTA.

in a few tens of minutes for an image of size $(512)^2 \times 33$ instead of several hours of computation with FISTA. Visually, a few iterations of IML FISTA are enough to obtain a good restoration, as shown in Figure 5.16 for the St Christopher HSI.

Discussion of the results. As said before, we expected IML FISTA Spat to perform better than IML FISTA Spec. Numerical experiments consistently show the opposite. A first explanation is that IML FISTA Spec averages the bands, which are all degraded differently. Therefore, if pixels are missing in a band, they may be present in its neighbors. This averaging operation can then be seen as a way to fill in the missing pixels. This is not the case for IML FISTA Spat, which averages the pixels of the same band. To check if this explanation was correct, we put to 0 the same 50% of pixels in each band and repeated the experiment. We obtained the same results that are displayed in Figure 5.15. These results suggest that fidelity to the fine level is more important than drastic dimension reduction to accelerate the optimization.

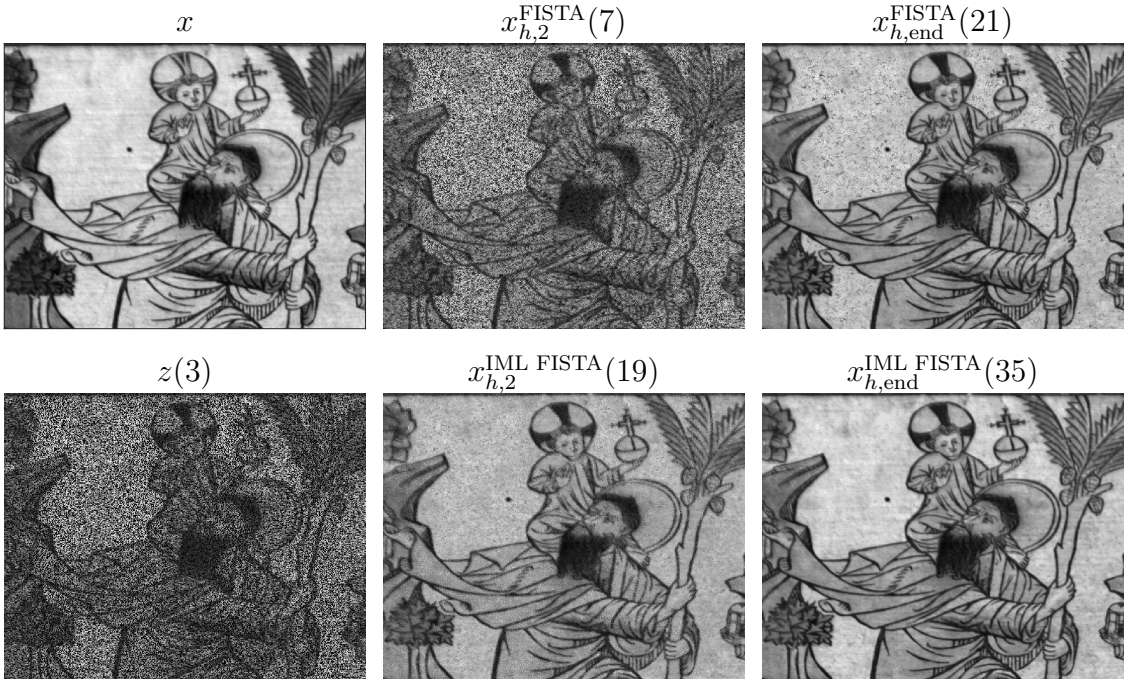

Figure 5.16: Inpainting ℓ_*-NLTV for the St-Christopher engraving hyperspectral image. Missing pixels 50%, σ(noise) = 0.01. On the left, objective function (normalized with initialization value) vs CPU time ($\times 10^4$ sec). We display the 15-th band of of the HSI for FISTA and IML FISTA after 2 iterations and at the end of the computation time budget (50 iterations of FISTA).

5.6.4 Application to deblurring and inpainting, combined

In this final experiment, we consider the restoration of a hyperspectral image degraded by a blur and missing pixels. This context is slightly harder than the inpainting one, as the blur occurs before the missing pixels. We will only consider the HSI of St Christopher for this experiment.

Data fidelity term. To perform the restoration of such images, we model the degradation as the combination of a Gaussian blur and a mask on the pixels (in this order). The parameters of the degradation are the following: missing pixels 50%; dimension(PSF) = 5; $\sigma(\text{PSF}) = 0.9$; $\sigma_{\text{noise}} = 0.01$.

Regularization term. We consider the same regularization as before, the structure tensor non-local TV penalization proposed in [33].

Information transfer operators. We have assessed that IML FISTA Spec was faster than IML FISTA Spat in the previous inpainting experiment. Thus, we will only consider IML FISTA Spec for this experiment, and the construction of the hierarchy is the same as in the previous experiment.

Multilevel parameters. The proposed multilevel algorithm has then 5 levels, and at the last level the HSI is of size $512 \times 512 \times 3$. The configuration remains the same as presented in previous experiments. In the previous study, we have also seen that $d = 0.5$ was a good trade-off between relaxing the necessary decrease of the proximity operator estimation's error and having a sufficient decrease of the objective function at each iteration with the inertia. We reuse this choice of d here.

Results. The evolution of the objective function and the reconstructed hyperspectral image of this experiment are displayed in Figure 5.17 and Figure 5.18. Essentially, the decrease of the objective function obtained by IML FISTA is faster than what it is obtained by FISTA on about 50 iterations while only calling **ML** twice.

5.7 Conclusion

In this chapter, we have shown that IML FISTA could vastly outperform FISTA, the state-of-the-art first order optimization method on the class of functions we consider, on a wide range of problems. We have shown that the algorithm is robust to the degradation context, whether it is deblurring or inpainting or a combination of the two. We also placed ourselves in really high dimensional settings with hyperspectral images. Under these settings, the available gains when using IML FISTA are massive.

Among its many advantages, IML FISTA provides good quality reconstructions faster than FISTA. This opens up a great opportunity to deal with problems of large dimension, especially when limited computational resources prevent convergence from being achieved systematically. In addition, this accelerated coarse approximation could play an important role in applications where image reconstruction is only a pre-processing task.

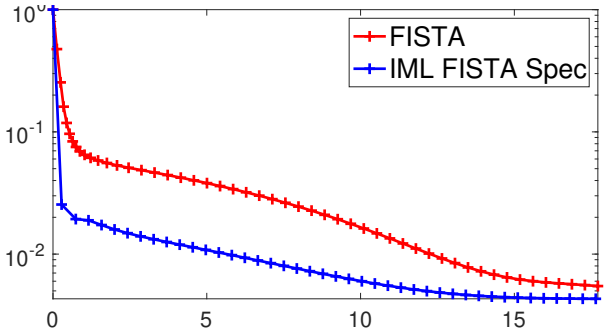

Figure 5.17: Blurring and inpainting ℓ_*-NLTV for the St-Christopher engraving hyperspectral image. Missing pixels 50%, $\dim(\text{PSF}) = 5$, $\sigma(\text{PSF}) = 0.9$, $\sigma(\text{noise}) = 0.01$. On the left, objective function (normalized with initialization value) vs CPU time ($\times 10^4$ sec). On the right, band 15 of the HSI for FISTA and IML FISTA after 2 iterations and at the end of the computation time budget (50 iterations of FISTA).

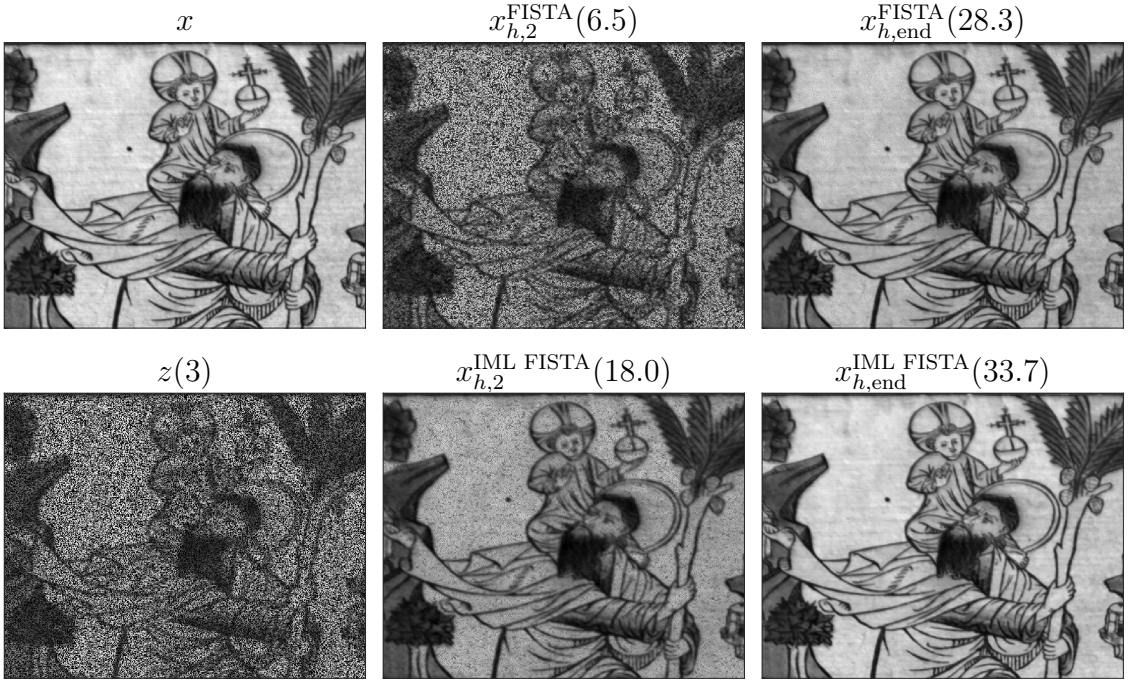

Figure 5.18: Blurring and inpainting ℓ_*-NLTV for the St-Christopher engraving hyperspectral image. Missing pixels 50%, $\dim(\text{PSF}) = 5$, $\sigma(\text{PSF}) = 0.9$, $\sigma(\text{noise}) = 0.01$. We display the 15-th band of of the HSI for FISTA and IML FISTA after 2 iterations and at the end of the computation time budget (50 iterations of FISTA).

At last, the higher the degradation, the better the performance of IML FISTA is relative to FISTA. This suggests that the algorithm is well suited to the most difficult problems. In order to convince the reader of the potential of IML FISTA, we will apply it to a real-world problem in the next chapter: the reconstruction of images obtained through radio-interferometry in astronomy.

Nonetheless, the performance of IML FISTA with respect to FISTA is not as good as that of a multigrid method with respect to a single grid method to solve PDEs. This is certainly due to the fact that FISTA still recovers the low frequency information of the solution first, therefore the coarse correction is not as efficient as it could be. But still, the gain is substantial, and the algorithm is very promising.

Chapter 6

IML FISTA: application to radio-interferometric imaging

In this chapter, we conclude with the applications of IML FISTA by tackling a radio-interferometric imaging problem. The design of our algorithm to tackle this problem is quite interesting, as the reduction of the dimension will not take place in the image directly but rather in a "dual" observation space.

The content of this chapter was partially published in the following paper [166]. It follows the same structure as the paper, with additional details: notably the generalization of the developed framework to more than 2-Levels; and additional illustrations.

This work was done in collaboration with Audrey Repetti and Yves Wiaux from Heriot-Watt University, Edinburgh, UK. Our contribution is built on the work done in the BASP Group[1] which includes some code, and domain specific knowledge.

6.1 Introduction

Motivated by the success of IML FISTA on relatively toy problems in the previous chapter, it is natural to wonder how our algorithm would perform on more realistic problems. In this chapter, we will apply IML FISTA to radio-interferometric imaging (RI), a problem that is of great interest in astronomy. This problem is particularly challenging due to the large amount of observations that are collected by radio telescopes, and thus is an excellent candidate to test the relevance of our approach.

Notations. In this chapter, we do not follow the convention of the RI literature, and of our own article [166], but rather those of the rest of the manuscript. For instance, the visibilities are denoted by z instead of y in the literature.

Organization of the chapter. We will first begin by a short presentation of the astronomy context and its challenges we want to address in Section 6.2. Then, we will present the radio-interferometric imaging problem and the state-of-the-art methods. Fi-

[1] https://basp.site.hw.ac.uk

Figure 6.1: The MeerKAT radio-interferometric array in South Africa. It consists of 64 antennas, and will be a part of the future Square Kilometer Array (SKA).

nally, we will present our IML FISTA algorithm whose construction is adapted for this problem.

6.2 Radio-interferometric imaging

Radio-interferometric (RI) imaging aims to reconstruct a sky brightness distribution from noisy observations in the Fourier space (named *visibilities*). These measurements are collected by what is called a radio-interferometer, an array[2] of antennas or dishes that combine the signal between pairs of antennas. In contrast, an optical telescope only has one dish. The last decades have seen the development of ever-increasing number of antennas for radio-interferometric arrays. From the Very Large Array (VLA) in New Mexico (USA) to the ongoing SKA in South Africa and Australia, the number of antennas has increased from a few and is expected to reach a few thousands in this decade (2020s). Typical array now possess dozens of antennas that are combined to sample the Fourier observations (see Figure 6.1 for an example of a radio-interferometric array). The distribution of the Fourier frequency samples is dictated by the position of the antennas used to probe the sky. The largest sampled spatial frequency is a direct function of the longest baseline, i.e. the distance between the two antennas that are the farthest apart. The larger the sampled frequencies, the higher the angular resolution of telescope and thus of the resulting image.

Classical diffraction theory says that the angular resolution θ of a telescope is limited by its diameter D according to the following relationship [9]:

$$\theta = 1.22\frac{\lambda}{D} \tag{6.1}$$

where λ is the wavelength of the observed signal. Radio-interferometers artificially increase the diameter D with the longest baseline $\tilde{D}$ between two antennas (see Figure 6.2). The number of antennas within an interferometer being naturally finite, this leads to under sampling the Fourier measurements [11]. Again this under sampling is a consequence of the array configuration. In Figure 6.4 you can see elliptic arcs crossing the Fourier space. Each one of them is the signal collected by a pair of two antennas across several

[2]We refer here to a network of several antennas.

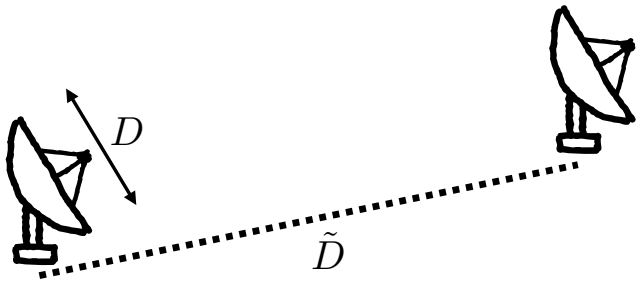

Figure 6.2: A schematic representation of a radio-interferometric baseline.

hours, the arc being the projection of the Earth rotation on the sky. The problem we are facing can be seen as an inpainting problem in the Fourier space.

Obtaining then the target image requires to deconvolve and denoise the Fourier measurements, otherwise a direct inverse Fourier transform would lead to a blurry image.

6.2.1 Imaging model

The problem we presented above can be mathematically formulated as follows. Our goal is to reconstruct the intensity image $\overline{x} \in \mathbb{R}^N$ from a set of measured complex *visibilities* $z \in \mathbb{C}^M$ acquired in the Fourier space. The corresponding discretized forward model can be formulated as follows [167]:

$$z = \mathbf{GFZ}\overline{x} + \epsilon := \mathbf{\Phi}\overline{x} + \epsilon \tag{6.2}$$

where

- $\mathbf{G} \in \mathbb{C}^{M \times d}$ is a sparse interpolation operator that maps the Fourier coefficients on a regular grid (inherited from the image) to the non-uniformly located visibilities,
- $\mathbf{F} \in \mathbb{C}^{d \times d}$ is the 2D Discrete Fourier Transform,
- and $\mathbf{Z} \in \mathbb{R}^{d \times N}$ is a zero-padding operator to properly map $\overline{x}$ for the convolution performed through the operator $\mathbf{G}$. The value of d is controlled by the size of the interpolation kernel of $\mathbf{G}$ [168].

The term ϵ stands for a centered white Gaussian noise[3]. Such perturbations have been investigated in [170], but more generally they can be taken into account with a careful calibration processing [9, 171, 172]. Typically, $M \gg N$ and is of the order of 10 millions visibilities (see [170] for more details).

This problem presents two main challenges. First, it is severely ill-posed, hence requiring advanced imaging techniques. Second, the size of the data streams coming from radio-telescopes is expected to be ever-increasing, thus raising the challenge of designing highly

[3]This modeling of the measurement operation is an approximation of the true measurement process [9, 169, 170]: we assume here a narrow field of view, white noise across all visibilities and no anisotropic perturbations.

scalable methods. We present now some well-known methods in the radio-interferometric imaging field.

6.2.2 Recovery techniques in radio-interferometry

CLEAN [173, 174] is the most used algorithm in RI imaging. It is similar to a matching pursuit method, but shows limitations when probing extended complex emissions, or for large numbers of point sources (e.g. single stars) [175]. From what is called the dirty image (the inverse Fourier transform of the measurement), CLEAN iteratively identifies the brightest point source and the corresponding point spread function is estimated from this source. The result of the convolution of the point spread function with the point sources is subtracted from the dirty image. This process is repeated until the residual image is below a certain threshold. There exist variations of this algorithm, and it has seen several improvements since its inception [176–178].

The CLEAN algorithm has really efficient implementation, and is often much faster than other existing methods. However, the imaging quality is not on par with these methods [44, 167, 179], which we will present now.

6.2.3 Variational approaches

Sparsity Averaging Reweighted Analysis (SARA). Penalized variational procedures have been proposed to improve the quality of the reconstructed images in a more general framework [9, 170]. Based on the works done to develop regularized imaging methods and algorithms to solve them, the state-of-the-art variational formulation in RI is Sparsity Averaging Reweighted Analysis [167], promoting average sparsity of the solution in a concatenation of bases through a reweighted-ℓ_1 procedure. Specifically, it aims at minimizing a log-sum prior by solving a sequence of weighted ℓ_1 problems [179–181]. Two SARA formulations have been proposed for imaging [44, 167, 181]: a constrained and an unconstrained one, uSARA, on which we will focus on in this chapter.

6.2.4 uSARA approach in a nutshell

Inverse problems of the form of (6.2) can be solved by defining iterations to

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \;\; F(x) := \underbrace{\frac{1}{2}\|\mathbf{\Phi}x - z\|^2}_{L(x)} + R(x), \tag{6.3}$$

where $R\colon \mathbb{R}^N \to]-\infty, +\infty]$ is a regularization function incorporating prior information on the target solution.

The uSARA problem corresponds to Equation (6.3), where R corresponds to a log-sum penalization to promote sparsity in the concatenation of the first eight Daubechies wavelet bases and the Dirac basis. Such a regularization is then handled using a reweighted ℓ_1 approach, that aims at solving a sequence of weighted ℓ_1 problems [44, 180, 181]. The resulting reweighting procedure, starting with $\mathbf{W}_0 = \lambda \mathrm{Id}$, can then be written as

$$\begin{aligned} &\text{for } i = 0, \dots, I \\ &\quad \left\lfloor \begin{aligned} &\widetilde{x}_{i+1} \in \underset{x \in \mathbb{R}^N}{\text{Argmin}}\, F_i(x) := L(x) + R(x, \mathbf{W}_i), \\ &\mathbf{W}_{i+1} = \text{Diag}\left(\lambda(\rho + |\mathbf{D}^* \widetilde{x}_{i+1}|)^{-1}\right), \end{aligned} \right. \end{aligned} \tag{6.4}$$

where $I > 0$ is the maximum number of reweighting steps,

$$R(x, \mathbf{W}_i) = \|\mathbf{W}_i \mathbf{D}^* x\|_1 + \iota_{\mathbb{R}^N_+}(x), \tag{6.5}$$

with $\mathbf{D}$ being the SARA dictionary (a concatenation of wavelet bases), $\mathbf{D}^*$ its adjoint, and $\iota_{\mathbb{R}^N_+}$ being the indicator function associated with the positive orthant, $\lambda > 0$ is a regularization parameter balancing the contribution of the regularization and the data fidelity terms, and $\rho > 0$ ensures stability of the method. As ρ tends to 0 the solution of the weighted ℓ_1-norm problem approaches that of the ℓ_0 pseudo-norm problem [181,182].

It has been shown in [179] that when solving approximately the minimization problem in (6.4) with a fixed number of forward-backward iterations, the sequence $(\widetilde{x}_i)_{i \in \mathbb{N}}$ converges to a critical point of F in (6.3).

Scaling to high-dimensional data. Many algorithms have been proposed to reduce the computational load induced by the number of visibilities. Most often, these algorithms consider only a subset of visibilities at each iteration. A first idea is to split the visibilities into blocks and to parallelize the action of $\mathbf{\Phi}$ block wise [169]. This parallelization can also be extended to the regularization term [183], this time by splitting the reconstructed image. This will not be considered in this chapter, as we wanted first to see if IML FISTA could provide acceleration in the simplest setting.

In [184], this approach was also extended to multi-spectral data with a modification of SARA to take into account spectral correlations. Other approaches solve approximations of the original problem in smaller dimensions, by selecting relevant visibilities in a sketching fashion (suitable random projections for instance) [185] or updating only with a fraction of the visibilities at each optimization step in an online manner [186]. Finally, acceleration schemes such as preconditioning strategies can be considered to better take into account the specific RI Fourier distribution [183].

Some of these techniques may be combined with the IML FISTA algorithm we propose in this chapter, and will be discussed at the end. We will now present our approach to tackle the uSARA problem with IML FISTA.

6.3 The multilevel framework for radio-interferometry

Without loss of generality, we again present the proposed multilevel strategy on a two-level case and will discuss when needed the extension to more levels. In this setting, we index the functions at the coarse level with subscript H, i.e. F_H, L_H and R_H, for F, L, and R, respectively.

Spirit of the method. Given an objective function at the fine level F, the goal of multilevel approaches is to build a coarse approximation F_H of F, which is cheaper to

optimize, to accelerate the minimization of F. Our multilevel algorithm consists of alternating iterations at the coarse level on F_H (**ML** steps) and at the fine level on F. Within the multilevel framework developed in Chapter 4, the minimization of (6.3) at the fine level can be computed using either forward-backward iterations or its accelerated inertial version FISTA. Then, the overall multilevel alternating procedure reads

$$\begin{aligned} &\text{for } k = 0, 1, \dots \\ &\quad \overline{x}_k = \mathbf{ML}(x_k) \\ &\quad x_{h,k+1} = \mathbf{FISTA}(\overline{x}_k). \end{aligned} \tag{6.6}$$

The crucial component of our multilevel strategy is the construction of a F_H in order to be consistent with F [90, 126].

Our main contribution in this chapter is to construct F_H exploiting properties of the RI problem, considering a coarse model in the data domain rather than in the image domain as done in Chapter 5, and leveraging the specific RI Fourier sub-sampling.

6.3.1 Proposed coarse model in data space

The usual approach to construct F_H consists in approximating F in a lower dimensional space. This would amount here to decrease the size of the image x and to formulate a similar optimization problem for a low resolution image (see Chapter 5). However, to take into account that the limiting factor in RI imaging is the large number of visibilities rather than the size of the sought image, we deviate from the classical multilevel scheme, and we construct a coarse model based on the following approximation of L:

$$(\forall x \in \mathbb{R}^N) \quad L_H(x) := \frac{1}{2}\|\mathbf{S}\mathbf{\Phi}x - \mathbf{S}z\|^2 \tag{6.7}$$

where $\mathbf{S}\colon \mathbb{C}^M \to \mathbb{C}^{M_H}$ is an operator reducing the data dimensionality M to a lower dimension $M_H < M$. Note that $L_H\colon \mathbb{R}^N \to \mathbb{R}$ is defined on the same space $\mathbb{R}^N$ as L, thus information transfer operators between levels, commonly used in standard multilevel algorithms [88–90, 126, 128], are not required in the proposed setting.

In general, choosing $\mathbf{S}$ to reduce computation complexity without sacrificing reconstruction accuracy is challenging, and some choices may lead to suboptimal reconstruction [185]. However, in our framework, L_H is only used to propel the minimization of the fine level objective function. We propose to sub-sample the Fourier coverage, that will enable preserving the reconstruction quality while reducing the computation complexity of the overall minimization method. This choice will be further discussed in Section 6.4.2.

Remark 13. *Formulation (6.7) is standard in the sketching literature, where* $\mathbf{S}$ *is typically a Gaussian random matrix so that minimizing (6.7) guarantees signal recovery [187]. Such a strategy has however many drawbacks for RI imaging that were investigated in [185]. Notably operator* $\mathbf{S\Phi}$ *is dense and thus computationally intensive to use in iterative optimization (i.e. each matrix vector product involving* $\mathbf{S\Phi}$ *is inefficient).*

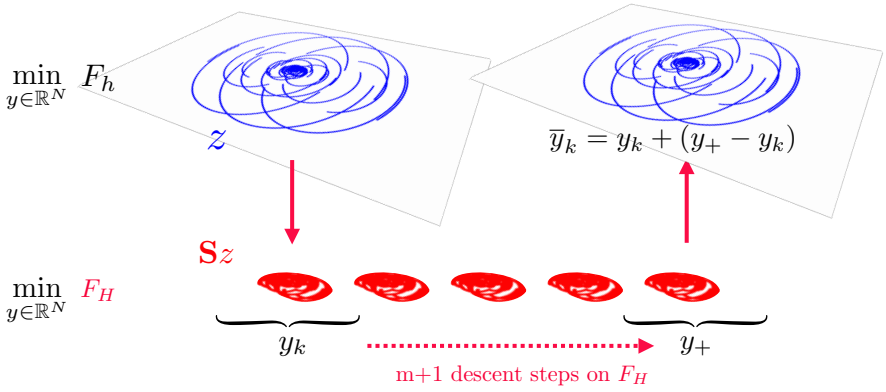

Figure 6.3: Scheme of a multilevel iteration in the radio-interferometric context. It encompasses the selection of a subset of the visibilities to define the coarse level; the definition of a coarse level function with these visibilities; several minimization steps at coarse level; and finally correction of the fine level iterate with the difference between the last and first coarse iterates. In this context there is no need for information transfer operators between levels: the reduction of the image dimension is done implicitly by selecting low frequencies in the Fourier space.

6.4 ML approach for uSARA acceleration

6.4.1 Proposed IML FISTA for uSARA

In the context of RI imaging we adapt IML FISTA for minimizing F_i at each reweighting step $i \in \{0, \dots, I\}$. Then, the proposed IML FISTA iterations for uSARA, read

$$\begin{aligned} &\text{Initialize } x_0 = z_0 = \widetilde{x}_i \text{ and for } k = 0, 1, \dots \\ &\left| \begin{aligned} &\overline{y}_k = \mathbf{ML}(y_k) \\ &x_{k+1} \approx \operatorname{prox}_{\tau R(\cdot, \mathbf{W}_i)} \left(\overline{y}_k - \tau \nabla L(\overline{y}_k)\right), \\ &y_{k+1} = x_{k+1} + \alpha_k (x_{k+1} - x_k) \end{aligned} \right. \\ &\text{Return } \widetilde{x}_{i+1} = x_{k+1}, \end{aligned} \tag{6.8}$$

where $\tau > 0$ and $(\alpha_k)_{k\in\mathbb{N}}$ are chosen according to [70], and the approximation errors on the proximal operator are assumed to be summable [70, 126]. The multilevel (**ML**) step consists of updating the variable y_k at certain iterations with a correction from coarse models to obtain a better update $\overline{y}_k$. The detailed version of this step and the variables involved are presented in Algorithm 9.

At iteration k of algorithm (6.8) the coarse objective function F_H is given for all $x \in \mathbb{R}^N$ by

$$F_H(x) = L_H(x) + R_{H,\gamma}(x, \mathbf{W}_i) + \langle v_{H,k}, x \rangle, \tag{6.9}$$

where

$$v_{H,k} = \nabla\Big(L + R_\gamma(\cdot, \mathbf{W}_i) - L_H - R_{H,\gamma}(\cdot, \mathbf{W}_i)\Big)(y_k). \tag{6.10}$$

Algorithm 9 ML step in (6.8)

if Coarse correction at iteration k **then**
  Set $\tau_H > 0$ and $\alpha_H > 0$ according to [70]
  $y_+ = \underbrace{(\mathrm{Id} - \tau_H \nabla F_H) \circ \ldots \circ (\mathrm{Id} - \tau_H \nabla F_H)}_{p \text{ gradient steps}}(y_k)$
  $\overline{y}_k = y_k + \alpha_H \left(y_+ - y_k\right)$
else
  $\overline{y}_k = y_k$
end if

In (6.10), $R_\gamma(\cdot, \mathbf{W}_i)$ corresponds to a smooth approximation of $R(\cdot, \mathbf{W}_i)$ with parameter $\gamma > 0$ [132, Definition 2.1], obtained using a Moreau-Yosida smoothing technique (see Chapters 4 and 5). The advantage of this technique is that the gradient has a closed form expression. Similarly, $R_{H,\gamma}$ is a smooth approximation of the coarse approximation R_H, built using the same technique. Note that a coarse model for F_H can be defined in the same manner, so that the coarse approximation could recursively benefit from its own coarse approximation.
In (6.9), $v_{H,k}$ imposes first-order coherence between the smoothed versions of the functions at fine and coarse levels. According to Theorem 5 (Chapter 4, Section 4.5.4), we then have the following theoretical guarantees:

Theorem 7. *Let $i \in \{1, \ldots, I\}$. Let $(x_k)_{k\in\mathbb{N}}$ and $(y_k)_{k\in\mathbb{N}}$ be sequences generated by algorithm (6.8). Assume that, for every $k \in \mathbb{N}$, the coarse model defined in Algorithm 9 decreases, i.e. $F_H(y_+) \leq F_H(y_k)$[a]. Then, the following assertions hold:*

1. *$\left(F(x_k) - F^*\right)_{k\in\mathbb{N}}$ is decreasing at a rate of $1/k^2$,*
2. *$(x_k)_{k\in\mathbb{N}}$ converges to a minimizer of F_i when $k \to \infty$.*

[a] This is ensured as soon as $\tau_H < \beta_H^{-1}$, where $\beta_H > 0$ is the Lipschitz constant of ∇F_H.

6.4.2 Algorithmic settings and implementation

Proximity operator computation. In (6.8), the proximity operator of $R(\cdot, \mathbf{W}_i)$ is defined, for every $x \in \mathbb{R}^N$, as

$$\mathrm{prox}_{\tau R(\cdot,\mathbf{W}_i)}(x) = \underset{u\in\mathbb{R}_+^N}{\arg\min} \frac{1}{2\tau}\|u - x\|^2 + \|\mathbf{W}_i\mathbf{D}^* u\|_1.$$

Since this proximity operator does not have a closed form expression, it can be computed with sub-iterations. In particular, the dual forward-backward algorithm proposed in [63] produces a sequence of feasible iterates converging to $\mathrm{prox}_{\tau R(\cdot,\mathbf{W}_i)}(x)$ [63, Theorem 3.7]. We detail now the procedure to compute this proximity operator.

Let $x \in \mathbb{R}^N$. $\mathbf{W}_i\mathbf{D}^* : \mathbb{R}^N \to \mathbb{R}^{9N}$ is a bounded non-zero linear operator such that the qualification condition

$$0 \in \ \mathrm{sri}\ (\mathbf{W}_i\mathbf{D}^*(\mathrm{dom}\iota_{\mathbb{R}_+^N}) - \mathrm{dom}\ \|\cdot\|_1) \tag{6.11}$$

holds (sri denotes the strong relative interior of a convex set). The problem of computing the proximity operator of the sum is (with $\tau = 1$ for simplicity) [63]

$$\min_{u \in \mathbb{R}^N} \iota_{\mathbb{R}^N_+}(u) + \|\mathbf{W}_i \mathbf{D}^* u\|_1 + \frac{1}{2}\|u - x\|^2 \tag{6.12}$$

Its dual problem is the following [63]:

$$\min_{v \in \mathbb{R}^{9N}} {}^1(\sigma_{\mathbb{R}^N_+})(x - \mathbf{W}_i \mathbf{D} v) + \|v\|_\infty \tag{6.13}$$

The notation ${}^1(\sigma_{\mathbb{R}^N_+})$ is to be understood as the Moreau envelope of parameter 1 of the conjugate of $\iota_{\mathbb{R}^N_+}$ (which is the support function of the set $\mathbb{R}^N_+$). This problem admits at least one solution, and every solution v^* is characterized by the inclusion [63, Proposition 3.3]

$$\mathbf{W}_i \mathbf{D}^* \Big(\mathrm{prox}_{\iota_{\mathbb{R}^N_+}}(x - \mathbf{W}_i \mathbf{D} v^*) \Big) \in \partial \|(v^*)\|_\infty. \tag{6.14}$$

Proposition 5. *Let v^* be a solution to Problem (6.13) and set*

$$z = \mathrm{prox}_{\iota_{\mathbb{R}^N_+}}(x - \mathbf{W}_i \mathbf{D} v^*). \tag{6.15}$$

Then z is the solution to Problem (6.12).

The solution of Problem (6.12) can be computed by Algorithm 6 [63]. This algorithm

Algorithm 10 Computation of the proximity operator. Let $(a_n)_{n\in\mathbb{N}}$ be a sequence in $\mathbb{R}^{9N}$ such that $\sum_{n\in\mathbb{N}} \|a_n\| < +\infty$ and let $(b_n)_{n\in\mathbb{N}}$ be a sequence in $\mathcal{H}$ such that $\sum_{n\in\mathbb{N}} \|b_n\| < +\infty$. Sequences $(u_n)_{n\in\mathbb{N}}$ and $(v_n)_{n\in\mathbb{N}}$ are generated by the following routines:

Require: $\eta \in]0, \min\{1, \|\mathbf{W}_i \mathbf{D}^*\|^{-2}\}[, v_0 \in \mathbb{R}^{9N}$

1: **for** $n = 0, 1, \ldots$ **do**
2: $\quad u_n = \mathrm{prox}_{\iota_{\mathbb{R}^N_+}}(x - \mathbf{W}_i \mathbf{D} v_n) + b_n$
3: $\quad \gamma_n \in [\eta, 2], \ \|\mathbf{W}_i \mathbf{D}^*\|^{-2} - \eta]$
4: $\quad \lambda_n \in [\eta, 1]$
5: $\quad v_{n+1} = v_n + \lambda_n \Big(\mathrm{prox}_{\gamma_n \|\cdot\|_\infty}(v_n + \gamma_n (\mathbf{W}_i \mathbf{D}^* u_n)) + a_n - v_n \Big).$
6: **end for**

is akin to a forward-backward algorithm and thus produces a decreasing sequence of dual objective function values.

Construction of S. To demonstrate the potential of the proposed IML FISTA for RI imaging, we choose $\mathbf{S}$ in (6.7) to select low-frequency coefficients in the Fourier $u - v$ coverage, and preserving the ellipsis arcs (i.e. corresponding to antenna pairs selecting low-frequency components). An example with a Fourier coverage simulated from a subset

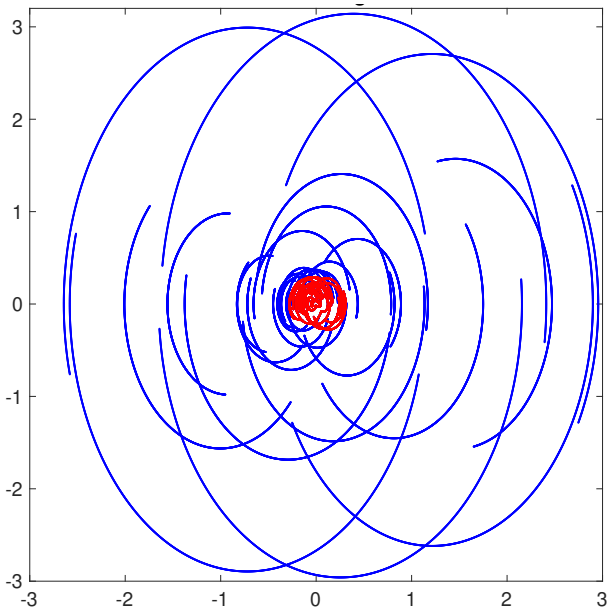

Figure 6.4: Fourier coverage for the fine (in blue) and coarse level (in red) when using the MeerKAT telescope [188].

of 64 antennas of the MeerKAT telescope [188] is displayed in Figure 6.4, where $\mathbf{S}$ selects $M/2$ coefficients (in red) out of the $M = 10,080,000$ total observations.

In other words we keep the visibilities produced by pair of antennas with the smallest distance to each others in the physical world. This choice is also based on the fact that most of the signal energy is usually concentrated around low frequencies [167] to accelerate the reconstruction of the image, a common idea in RI imaging [183].

Formally $\mathbf{S}$ is a sub-sampling operator that selects a subset $J \subseteq \{1, \dots, m\}$ of the available visibilities. We then construct $\mathbf{\Phi}_H = \mathbf{S}\mathbf{\Phi}$ to map the DFT of the image x to $\mathbf{S}y$.

Choice of coarse model. We choose $R_{H,\gamma} = 0$, i.e. the coarse level is not regularized explicitly. This choice is due to the fact that through $\mathbf{S}$ we are only working with low-frequencies, and we observed that adding a coarse regularization in this case was not making a quantitative difference in preliminary numerical experiments[4]. Thus, the coarse objective function (6.9) boils down to

$$(\forall x \in \mathbb{R}^N) \quad F_H(x) := \frac{1}{2}\|\mathbf{S}\mathbf{\Phi}x - \mathbf{S}z\|^2 + \langle v_H, x\rangle. \tag{6.16}$$

Hence, the coarse model is still guided by the fine level regularization through $v_{H,k}$.

Regarding the smoothing of $R(\cdot, \mathbf{W}_i)$ in (6.10), we choose to only smooth the SARA weighted-ℓ_1 regularization without enforcing the first order coherence with respect to $\iota_{\mathbb{R}^N_+}(\cdot)$. Moreover, one can think of **ML** steps as some kind of "gradient" steps, before applying the proximity operator that enforces the feasibility. In practice, we have not observed unfeasible coarse iterates.

[4]We did not try to remove the regularization in the experiments from Chapter 5, it may be another way of improving the performance of IML FISTA in these applications.

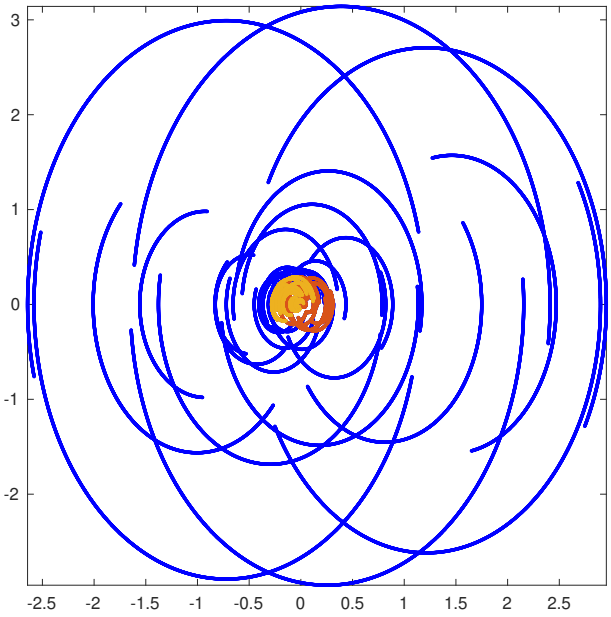

Figure 6.5: Fourier coverage for level 3 (fine) (in blue) and coarse level 2 (in red) and 3 (in orange) when using the MeerKAT telescope [188].

Multilevel setup. For all the performed experiments, our multilevel algorithm will consist of 3 levels. Each coarse level has half of the available measurements of its corresponding fine level. At each level we perform $p = 5$ iterations of gradient descent in algorithm 9 with F_H given in (6.16). The coarse models are called every iteration, i.e., we never compute fine level updates alone.

6.4.3 Generalization to more than two levels

It may not be straightforward for the reader to see how one could generalize the proposed method to more than two levels. In fact, to construct more than two levels we will repeat the same procedure as to define the coarse level in the previous paragraph and construct a series of $(\mathbf{S}_\ell)_{2\leq\ell\leq L}$ that select visibilities of level ℓ to define the coarse level of level $\ell-1$. The coarse model of level $\ell-1$ will then be defined as in (6.16) with $\mathbf{S} = \mathbf{S}_L$.

$$(\forall 2 \leq \ell \leq L)\ (\forall x \in \mathbb{R}^N) \quad F_{\ell-1}(x) := \frac{1}{2}\| \left(\prod_{j=\ell}^{L} \mathbf{S}_i\right) \mathbf{\Phi} x - \left(\prod_{j=\ell}^{L} \mathbf{S}_i\right) z\|^2 + \langle v_H, x\rangle. \quad (6.17)$$

The computation of $\left(\prod_{j=\ell}^{L} \mathbf{S}_i\right) \mathbf{\Phi}$ and of $\left(\prod_{j=\ell}^{L} \mathbf{S}_i\right) z$ is done once at the start of the optimization.

The multilevel algorithm will then consist of L levels, each level having half of the available measurements of its corresponding fine level. You can see in Figure 6.5 the selection of visibilities when using 3 levels.

By increasing the number of levels in this case of coverage, one can target lower frequencies, which contain most of the signal here (note that this may not be true for other coverages).

6.5 Numerical experiments

6.5.1 Dataset

We use a subset of 64 antennas from the MeerKAT array [188]. Each antenna pair acquires $5,000$ visibilities, leading to a total of $M = 10,080,000$ observations (see Figure 6.4 for the resulting Fourier coverage). In our simulations, we use a simulated image of the M31 galaxy[5] of dimension $n = 512 \times 512$, provided by the BASP group. The measurements are obtained as per equation (6.2), where $\epsilon \in \mathbb{C}^m$ is a realization of a centered white Gaussian noise with variance $\sigma = 0.007$, so that the input Signal-to-Noise-Ratio (SNR) is equal to 19 dB in the visibility domain.

6.5.2 Minimization comparison without reweighting

In this section we will compare three optimization methods for solving Equation (6.3): FB, FISTA, and IML FISTA. Each algorithm is given a budget of CPU time to reach the best reconstruction (λ is chosen via grid search). Our main goal is to demonstrate that IML FISTA is faster than FISTA to solve this problem. First and foremost we are interested in the quality of the reconstruction, so we will plot two criteria to validate the performances of our algorithm: the objective function and the SNR evolution with respect to the CPU time, in Figures 6.6 left and middle, respectively.

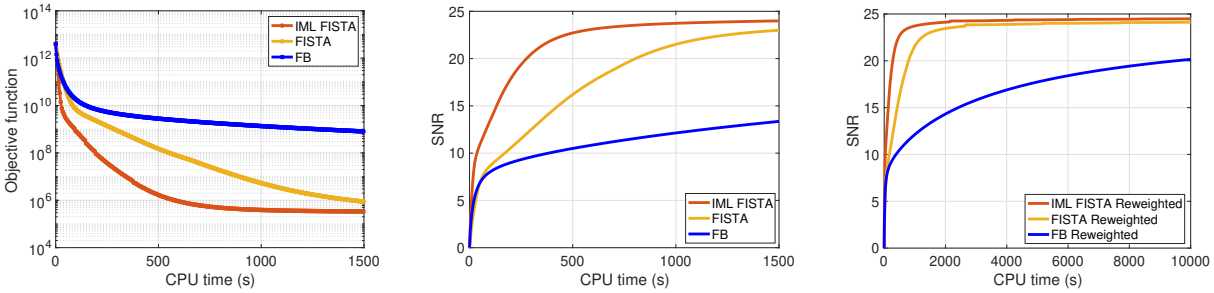

Figure 6.6: Evolution of the objective function values **(left)**, of the SNR **(middle)** of the iterates produced by FB algorithm, FISTA and IML FISTA with respect to the CPU time when solving Problem (6.3) (each algorithm had a CPU time budget of 1500 seconds). Evolution of the SNR for the three algorithms when we involve the complete reweighting procedure **(right)** (CPU time budget of 10000 seconds).

As one can see IML FISTA outperforms both FISTA and FB algorithms for a single round of convex optimization. We further provide reconstructions obtained with the three methods for visual inspection in Figure 6.7, at given CPU computation times $\{\approx 300\text{s}, \approx 500\text{s}, \approx 800\text{s}, \approx 1,100\text{s}, \approx 1,500\text{s}\}$.

6.5.3 Minimization comparison for uSARA

We now focus on solving the complete uSARA problem. As we solve a sequence of optimization problems (6.4) that will be different for each optimization method, the easiest

[5]Image available at: https://casaguides.nrao.edu/index.php?title=Sim_Inputs.

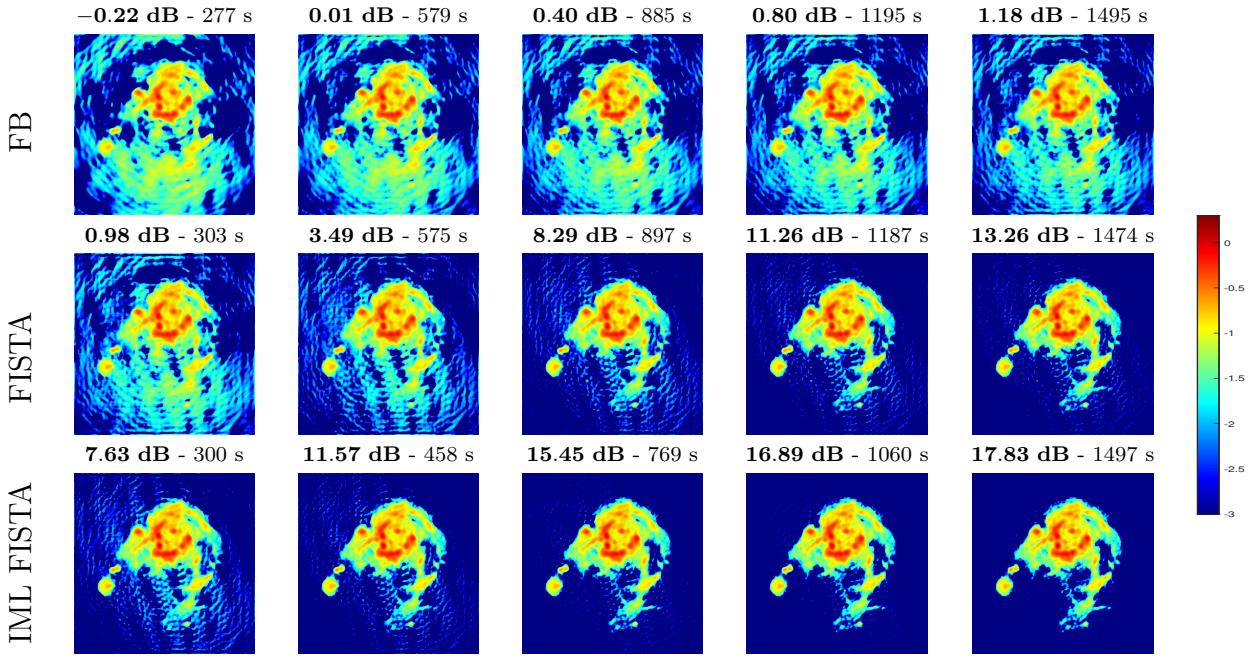

Figure 6.7: Reconstruction in log scale of a region of the M31 galaxy by FB (top row) FISTA (middle row) and IML FISTA (bottom row) at equivalent CPU times. The legend on top of each thumbnail reads as follows: **log SNR** in dB - CPU time in seconds. log SNR $= \mathrm{SNR}(\log_{10}(10^3 x + 1)/3, \log_{10}(10^3 x_{\text{truth}} + 1)/3)$.

way to evaluate each method is to only compare FB, FISTA and IML FISTA on the SNR evolution with respect to the CPU time computation. The results are shown in Figure 6.6-right. One can see that at each "reweighting" step a small jump in the SNR of the iterates occurs (for both FISTA and IML FISTA) due to the reset of the inertia parameters (for convergence reasons [70]). With the given coverage we can only slightly improve the SNR of the reconstruction, but nevertheless IML FISTA reaches an upper bound faster than FISTA.

6.6 Conclusion

Conclusion. In this chapter we proposed a multilevel approach for solving the uSARA problem in RI imaging, where the coarse level enables working with low-dimensional data, while the fine model ensures consistency with the full data and promotes averaging sparsity. We have also integrated the resulting IML FISTA iterations, within a reweighting framework, further enhancing sparsity. We have shown through simulations on RI imaging that the proposed IML FISTA leads to impressive acceleration with respect to FB to solve uSARA by exploiting approximations in the observation space of the problem. Our method shows promising results on simulations when integrated within the reweighting procedure. It remains to conduct more extensive experiments to further explore the benefits and possible gains of the procedure, as radio-interferometric imaging is a vast domain where numerous configurations of measurements can happen depending on the astronomical object to observe.

Perspectives. The proposed multilevel acceleration for uSARA being very promising, we have identified future research directions to better assess its potential for RI imaging.

On the one hand, there exist approaches to efficiently handle high-dimensional data based on a parallel implementation of the measurement operator $\mathbf{\Phi}$ [169]. Our method could be coupled with such a parallelization strategy to benefit at the same time from the dimensionality reduction at the coarse levels and from an efficient parallel implementation of $\mathbf{\Phi}$ at the fine levels.

This parallelization strategy exploits the distribution of the measurements across the Fourier space, and the structure of the interpolation kernel to construct a parallel implementation of the measurement operator. One could exploit in similar fashion this construction to select the measurements at coarse level, so that the subsequent measurement operator enjoys the same parallelization properties.

Also, preconditioning strategies enabling natural weighting (leveraging the local density of the Fourier sampling) could be considered for comparison and/or further acceleration of the proposed method [183].

Moreover, sophisticated coarse models for RI could be investigated to potentially improve the results (for instance sketching approaches [185]).

Furthermore, connections of multilevel approaches with CLEAN algorithm [173] and its learned version R2D2 [189] could be studied. Both methods are built on major-minor cycles reminiscent of matching pursuit. During the minor cycles, an approximate data term is used, ultimately enabling a much smaller number of major cycles (requiring passing through the full data). This is akin to the proposed multilevel method.

On the other hand, a few theoretical research directions could be pursued. Leveraging approximation theory as in [179], the global convergence of the multilevel strategy within a reweighting framework could be studied. The multilevel framework, as we have seen at the end of Chapter 4, could also be extended to primal-dual algorithms, to enable solving the constrained formulation of SARA, which in this RI context yields better reconstruction quality. At this point, multilevel primal-dual algorithms are not well enough understood to tackle such imaging problem (see Appendix A.3.2), but it would nevertheless be interesting to investigate this direction.

Part III

Multilevel optimization: a new perspective

Chapter 7

Multilevel algorithms from a block-coordinate descent point of view

In this final part of the manuscript we revisit the theoretical construction of multilevel algorithm from the point of view of coordinate descent algorithms. This point of view emerged from revisiting the very first image restoration problem we tackled during my PhD: the ℓ_1-wavelet-regularized deblurring problem. We establish an equivalence between IML FB and a block-coordinate descent (BCD) algorithm having a hierarchical – in a multilevel sense – selection of the blocks to update. On the way, we prove the convergence of this BCD algorithm in a non-convex, non-smooth setting. We leave for later works the study of the equivalence between IML FISTA and inertial BCD algorithms [190].

This work was done in collaboration with L. Briceño-Arias from Universidad Técnica Federico Santa María, Chile.

7.1 Introduction

Our motivation. We have seen in Chapter 4 how to construct a convergent multilevel algorithm for non-smooth optimization. Then we showed in Chapter 5 and in Chapter 6 that this algorithm had great potential to accelerate the optimization of a wide range of large-scale imaging problems.

This acceleration has come with a what I would call a pre-computation cost: we worked quite a lot to identify a robust construction for the algorithm, that could be adapted to each one of the considered applications. Such search takes a good amount of time and knowledge about the problem at hand. Given that multilevel algorithms in the literature do seem to accelerate the solution of the associated optimization problem (to name a few [88–90, 103, 107, 108, 112]), one can imagine that the authors of these papers followed a similar path.

Therefore, I was strongly motivated to find a way to construct *ad hoc* multilevel algorithms, i.e., that could help us identify quickly good constructions. This is what this last part is about. The reader should keep in mind that the findings of this chapter came after the work presented in Chapter 4, 5, and 6.

Multilevel algorithm are block-coordinate descent algorithms. In the present chapter, we propose to construct an algorithm that fits at the same time the multilevel formalism[1] and the BCD formalism, so that the analysis of the multilevel algorithm would be done using BCD tools. Such connection was briefly mentioned in [112, Section 2.4], but their proposed multilevel algorithm did not follow the classic rules of multigrid/multilevel (see Chapter 3 and Chapter 4).

The analogy we draw between the block-coordinate algorithm and the multilevel algorithm goes both ways. We will show that the two algorithms are in fact the same for this problem, and that the block-coordinate point of view can be used to construct a multilevel algorithm. Then we will show that adopting multilevel precepts can help choose efficient update rules for block-coordinate algorithms and the block-coordinate point of view can give additional theoretical guarantees to multilevel algorithms.

To draw this analogy, we will construct a new BCD algorithm, able to handle non-convex, non-smooth optimization problems.

Organization of the chapter. This chapter is organized as follows. We first present the multilevel algorithm we want to analyze in the simplest setting, so that the reader can understand where we are heading: we will solve a deblurring problem with ℓ_1-wavelet regularization with a 2-levels algorithm and a 2-blocks algorithm.

Then we will present extensively the block-coordinate context, before providing our own BCD algorithm. The rest of the chapter will be dedicated to the study of this new BCD algorithm, and the analysis of multilevel algorithms through this lens.

We will end this chapter with some numerical experiments to confirm our theoretical findings.

7.2 A compelling example

In this section, for the sake of clarity, we recall some key facts about multiresolution analysis needed to understand the use of wavelet made in this chapter; then present the construction of the block algorithm, and of the multilevel algorithm for respectively two blocks and two levels. This presentation will allow us to highlight the parallels between the two approach on the simplest setting.

7.2.1 Key facts about multiresolution analysis

Let $x \in \Omega \subset \mathbb{R}^2$ be an image, such that $x \in L_2(\Omega)$. We can decompose $L^2(\Omega)$ into a sum of two spaces, that of approximation coefficients V_J and that of detail coefficients $V_J^\perp$ at resolution $J \in \mathbb{N}$ [4]:

$$L_2(\Omega) = V_J \oplus V_J^\perp. \tag{7.1}$$

We will assume that x lives exclusively in V_J in the following, i.e., x has 2^{2^J} pixels. V_J can be decomposed into subspaces V_{J-1} and W_{J-1}, where V_{J-1} is the space of approximation coefficients at resolution $J-1$ and W_{J-1} the space of detail coefficients at resolution $J-1$. x is then decomposed exactly as:

$$x = \Pi^*_{V_{J-1}} a_{J-1} + \Pi^*_{W_{J-1}} d_{J-1}. \tag{7.2}$$

[1] Following the principles presented in Chapter 4.

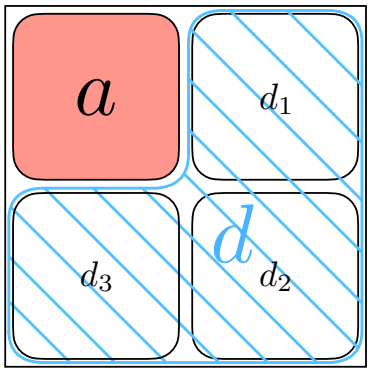

Figure 7.1: Decomposition on two levels of an image x with a wavelet transform. We regroup the detail coefficients d_1, d_2 and d_3 into one single block d to simplify the presentation of our two level or two block proximal gradient descent algorithms.

a_{J-1} (resp. d_{J-1}) are the approximation (resp. detail) coefficients at resolution $J-1$. Note that $\Pi_{V_{J-1}}\Pi^*_{V_{J-1}} = \mathrm{Id}_{V_{J-1}}$ and $\Pi_{W_{J-1}}\Pi^*_{W_{J-1}} = \mathrm{Id}_{W_{J-1}}$ where $\mathrm{Id}_{V_{J-1}}$ and $\mathrm{Id}_{W_{J-1}}$ are the identity operators on V_{J-1} and W_{J-1} respectively. Also, we won't take into account in our presentation the fact that wavelet transform of an image yields one block of approximation coefficients and three blocks of detail coefficients. The detail coefficients will be grouped together in a single block belonging to W_{J-1} (which will therefore be three times as big as the approximation block), see Figure 7.1.

7.2.2 Wavelet deblurring: a block-multilevel algorithm

Assume that we can decompose our 2^{2^J} pixels image x into two independent components such that $x = \Pi^*_{V_{J-1}} a_{J-1} + \Pi^*_{W_{J-1}} d_{J-1}$ with $a_{J-1} \in V_{J-1}$ and $d_{J-1} \in W_{J-1}$. The index $J-1$ is dropped in the following for simplicity. The detail coefficients are grouped into one block d as stated in the previous section. We want to minimize the following objective function:

$$\underset{x \in V_J}{\mathrm{Argmin}}\, F(x) = \frac{1}{2}\|\mathrm{A}x - z\|_2^2 + \|\mathrm{D}x\|_1 \tag{7.3}$$

where A is a bounded linear operator, D is the wavelet transform of x on one level, and λ is multivalued to penalize differently the approximation and detail coefficients. We can rewrite this classical ℓ_1-wavelet penalized least-squares problem with the wavelet decomposition of x:

$$\underset{a\in V, d\in W}{\mathrm{Argmin}}\, \Psi(a,d) = \frac{1}{2}\|\mathrm{A}\left(\Pi^*_V a + \Pi^*_W d\right) - z\|^2 + \lambda_a\|a\|_1 + \lambda_d\|d\|_1 \tag{7.4}$$

Minimizing Ψ with respect to a and d is perfectly equivalent to minimizing F with respect to x, as we recover the solution of Problem (7.3), $\widehat{x}$, with the solution of Problem (7.4)

$$\widehat{x} = \Pi^*_V \widehat{a} + \Pi^*_W \widehat{d}. \tag{7.5}$$

Two block-coordinate proximal gradient descent. To minimize Ψ, our first strategy is using a block-coordinate approach starting from a_0, d_0, and with $0 < \tau < 2/\|\mathrm{A}^*\mathrm{A}\|$

and $\varepsilon_{a,n}, \varepsilon_{d,n} \in \{0,1\}$.

$$\begin{array}{l}\text{for } n = 0, 1, \dots \\ \left| \begin{array}{l} a_{n+1} = a_n + \varepsilon_{a,n} \left(\mathrm{prox}_{\tau\lambda_a \|\cdot\|_1} \left(a_n - \tau \Pi_V \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* a_n + \Pi_W^* d_n \right) - z \right) \right) - a_n \right) \\ d_{n+1} = d_n + \varepsilon_{d,n} \left(\mathrm{prox}_{\tau\lambda_d \|\cdot\|_1} \left(d_n - \tau \Pi_W \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* a_n + \Pi_W^* d_n \right) - z \right) \right) - d_n \right) \end{array} \right. \end{array} \tag{7.6}$$

Setting $(\varepsilon_{a,n}, \varepsilon_{d,n}) = (1,1)$ for all n leads to the standard forward-backward algorithm. A cyclic coordinate descent algorithm consists in setting alternatively one of $\varepsilon_{a,n}, \varepsilon_{d,n}$ to 1. This could also be set at random, provided that $\mathsf{P}\left[(\varepsilon_{a,n}, \varepsilon_{d,n}) = (0,0)\right] = 0$.

In a typical multilevel fashion, we would alternate between updating a alone, then a and d together (i.e., if $(\varepsilon_{a,n}, \varepsilon_{d,n}) = (1,0)$ then $(\varepsilon_{a,n+1}, \varepsilon_{d,n+1}) = (1,1)$).

In all cases, the convergence of the resulting algorithm to a minimizer of Ψ is guaranteed under the assumption that Ψ is convex and that the step size τ is chosen properly. We leave the proof for the general case.

Two-level proximal gradient descent. We present now the construction of a two-level algorithm to minimize Ψ. We will denote by Ψ_H the coarse level function.

Given the structure of the problem, it is natural to define Ψ_H in the approximation space V. Consequently, and following the rule and guidelines established in Chapter 4, the information transfer operator I_h^H is the projection Π_V onto V, and the prolongation operator I_H^h is directly Π_V^*. We also project z to V.

The coarse linear operator A_H can be then naturally defined as the Galerkin approximation of A (which is applied to x) so that $\mathrm{A}_H = \Pi_V \mathrm{A} \Pi_V^*$. The coarse model is thus defined as:

$$\Psi_H(a) = \frac{1}{2} \|\mathrm{A}_H a - \Pi_V z\|_2^2 + \lambda_a \|a\|_1 + \langle v_H, a \rangle \tag{7.7}$$

where v_H enforces the first order coherence between two smoothed version (Chapter 4, Definition 25) of Ψ and Ψ_H (Chapter 4, Definition 26 and Lemma 8)

$$v_H = \Pi_V \nabla \Psi_\mu(a_0, d_0) - \nabla(\|\mathrm{A}_H \cdot\|_2^2 + \|\cdot\|_{1,\mu})(a_0). \tag{7.8}$$

For simplicity, we denote $\|\cdot\|_{1,\mu}$ the $\mu > 0$-smoothed ℓ_1-norm (according to the principles of Chapter 4, Section 4.2.2) and suppose that we compute only one coarse iteration before going back to the fine level. This iteration will yield $a_{n+1/2}$ from a_n. The coarse level being non-smooth, we will use a proximal gradient step to decrease it.

Accordingly, the two-level proximal gradient algorithm is then defined as:

$$\begin{array}{l}\text{for } n = 0, 1, \dots \\ \left| \begin{array}{l} a_{n+1/2} = \mathrm{prox}_{\tau\lambda_a \|\cdot\|_1} \left(a_n - \tau \mathrm{A}_H^* \left(\mathrm{A}_H a_n - \Pi_V z \right) - \tau v_H \right) \\ a_{n+1} = \mathrm{prox}_{\tau\lambda_a \|\cdot\|_1} \left(a_{n+1/2} - \tau \Pi_V \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* a_{n+1/2} + \Pi_W^* d_n \right) - z \right) \right) \\ d_{n+1} = \mathrm{prox}_{\tau\lambda_d \|\cdot\|_1} \left(d_n - \tau \Pi_W \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* a_{n+1/2} + \Pi_W^* d_n \right) - z \right) \right) \end{array} \right. \end{array} \tag{7.9}$$

The fact that Algorithms 7.6 and 7.9 are the same algorithm is not obvious at first sight. We will show in the following that this is indeed the case. After summarizing our assumptions, we will compute the first order coherence term explicitly.

Assumption 3. *We assume that:*

(i) the information transfer operator is the projection Π_V onto V;

(ii) in the definition of v_H, the fine and coarse models are smoothed with the same smoothing technique, with the same smoothing parameter $\mu > 0$;

(iii) Ψ and Ψ_H are first order coherent with respect to their smoothed versions (Definition 26).

Lemma 17. *Suppose that Assumption 3 holds. The first order coherence term v_H in Equation (7.7) at point (a_0, d_0) is given by:*

$$v_H = \Pi_V \mathrm{A}^* \left(\mathrm{A}\Pi_W^* d_0 - \Pi_W^* \Pi_W z\right). \tag{7.10}$$

The first order coherence sends to the coarse level the contribution of the detail coefficients to the gradient of the data fidelity term.

Proof. By definition of first order coherence between smoothed functions, we have:

$$v_H = \Pi_V \nabla \Psi_\mu(a_0, d_0) - \nabla \Psi_{H,\mu}(a_0). \tag{7.11}$$

The term on the right is a simple computation of the gradient of the coarse model:

$$\begin{aligned}
\nabla \Psi_{H,\mu}(a_0) &= \nabla \left(\frac{1}{2}\|\mathrm{A}_H a_0 - \Pi_V z\|_2^2 + \lambda_a \|a_0\|_{1,\mu}\right) \\
&= \mathrm{A}_H^* \left(\mathrm{A}_H a_0 - \Pi_V z\right) + \lambda_a \nabla_a \left(\|\cdot\|_{1,\mu}\right)(a_0) \\
&= \Pi_V \mathrm{A}^* \left(\mathrm{A}\Pi_V^* a_0 - \Pi_V^* \Pi_V z\right) + \lambda_a \nabla_a \left(\|\cdot\|_{1,\mu}\right)(a_0)
\end{aligned} \tag{7.12}$$

The term on the left in Equation (7.10) is a bit more involved. We have:

$$\begin{aligned}
\nabla \Psi_\mu(a_0, d_0) &= \nabla \left(\frac{1}{2}\|\mathrm{A}\left(\Pi_V^* a_0 + \Pi_W^* d_0\right) - z\|_2^2 + \lambda_a \|a_0\|_{1,\mu} + \lambda_d \|d_0\|_{1,\mu}\right) \\
&= \mathrm{A}^* \left(\mathrm{A}\left(\Pi_V^* a_0 + \Pi_W^* d_0\right) - z\right) + \begin{bmatrix} \lambda_a \nabla_a \left(\|\cdot\|_{1,\mu}\right)(a_0) \\ \lambda_d \nabla_d \left(\|\cdot\|_{1,\mu}\right)(d_0) \end{bmatrix}
\end{aligned} \tag{7.13}$$

And now:

$$\begin{aligned}
v_H = \Pi_V &\left(\mathrm{A}^* \left(\mathrm{A}\left(\Pi_V^* a_0 + \Pi_W^* d_0\right) - z\right) + \begin{bmatrix} \lambda_a \nabla_a \left(\|\cdot\|_{1,\mu}\right)(a_0) \\ \lambda_d \nabla_d \left(\|\cdot\|_{1,\mu}\right)(d_0) \end{bmatrix}\right) \\
&- \Pi_V \mathrm{A}^* \left(\mathrm{A}\Pi_V^* a_0 - \Pi_V^* \Pi_V z\right) + \lambda_a \nabla_a \left(\|\cdot\|_{1,\mu}\right)(a_0)
\end{aligned}$$

As V and W are orthogonal to each other, $\Pi_V \left(\lambda_d \nabla \left(\|\cdot\|_{1,\mu}\right)(d_0)\right) = 0$ and thus:

$$v_H = \Pi_V \mathrm{A}^* \left(\mathrm{A}\Pi_W^* d_0 - \Pi_W^* \Pi_W z\right),$$

where we used that

$$z - \Pi_V^* \Pi_V z = \Pi_W^* \Pi_W z.$$

□

With this in mind, let us compute explicitly the proximal gradient step at coarse level at iteration n, specifying a_0 and d_0 as a_n and d_n in Equation (7.10):

$$\begin{aligned} a_{n+1/2} &= \operatorname{prox}_{\tau\lambda_a\|\cdot\|_1}\left(a_n - \tau \mathrm{A}_H^*\left(\mathrm{A}_H a_n - \Pi_V z\right) - \tau v_H\right) \\ &= \operatorname{prox}_{\tau\lambda_a\|\cdot\|_1}\left(a_n - \tau \Pi_V \mathrm{A}^*\left(\mathrm{A}\Pi_V^* a_n - z + \mathrm{A}\Pi_W^* d_n\right)\right) \\ &= \operatorname{prox}_{\tau\lambda_a\|\cdot\|_1}\left(a_n - \tau \Pi_V \mathrm{A}^*\left(\mathrm{A}\left(\Pi_V^* a_n + \Pi_W^* d_n\right) - z\right)\right) \end{aligned} \tag{7.14}$$

Therefore, the proximal gradient step at coarse level is exactly equal to a proximal gradient step at fine level, with respect to the approximation coefficients. We summarize the consequence of this result in the following lemma:

Lemma 18. *The two-level algorithm defined in Equation (7.9) is equivalent to the block-coordinate algorithm defined in Equation (7.6) when choosing $\varepsilon_{a,2n} = 1$ and $\varepsilon_{d,2n} = 0$, then $\varepsilon_{a,2n+1} = 1$ and $\varepsilon_{d,2n+1} = 1$ for all n.*

A direct consequence of this lemma is that **the two-level algorithm can be analyzed as a block-coordinate descent algorithm with specific update rules**.

Seeking convergence guarantees for a multilevel algorithm can now be deferred to seeking convergence guarantees for BCD algorithms, which are much more studied in the literature (see Section 7.3.2 for references). Moreover, this point of view gives us some precious insights on the construction of multilevel algorithms. We will present them in full in Section 7.5.

In the next sections, we will present a convergent block-coordinate descent algorithm that allows the type of updates described in Lemma 18. Notably, the difficulty lies in allowing parallel updates of the blocks, and non-independence of the choice of the blocks to update from one iteration to the other. The block update scheme we employ to emulate a multilevel algorithm with a block-coordinate descent algorithm is presented in Figure 7.2. It displays the parallel updates, the non-independence and the notion of cycle, which is crucial for the convergence of our algorithm.

The presentation of our BCD algorithm will adopt the point of view of the rest of the literature on the subject, which may be quite different from that of the multilevel algorithms. We will include some anchor points along the way so that we don't lose sight of the connection we are trying to draw between multilevel and BCD approaches.

7.3 Block-coordinate descent methods: quick overview

In this section, we introduce the relevant literature and technical background about block-coordinate descent algorithm. The reader familiar with this context can skip this section and go straight to Section 7.4.

Many problems in machine learning or signal and image processing, consist in minimizing a sum of two functions, one encoding a fidelity with respect to some observation (data) and the other encoding some prior knowledge about the parameters (e.g. an image) to estimate. The associated **non-smooth and non-convex** optimization problem is

$$\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}\in\mathcal{H}} \Psi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}), \tag{7.15}$$

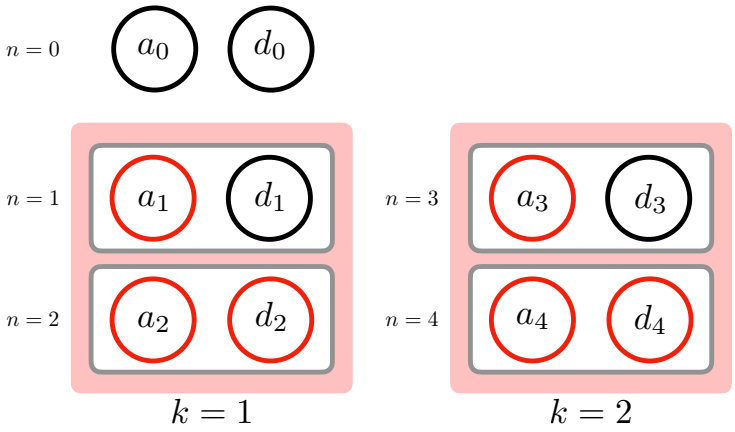

Figure 7.2: Update scheme of the two block-coordinate descent algorithm. The blocks are updated in a cyclic fashion, first with the approximation block updated alone (a_1 in red), then the approximation and detail blocks updated together (a_2 and d_2 in red). We represent two cycles in this figure $k = 1$ and $k = 2$ for a total of $n = 4$ iterations. This notion of cycle will be crucial for our block-coordinate descent algorithm.

where $f : \mathcal{H} \mapsto (-\infty, +\infty]$ and $g : \mathcal{H} \mapsto (-\infty, +\infty]$.

We will consider the real finite dimensional Hilbert space $\mathcal{H} := \bigoplus_{\ell=1}^{L} \mathcal{H}_\ell$, as the direct sum of L separable Hilbert spaces $(\mathcal{H}_\ell)_{1\leq\ell\leq L}$, meaning that all $\mathbf{x} \in \mathcal{H}$ can be decomposed into L blocks such that $\mathbf{x} = (x_1, \ldots, x_L)$ with $x_\ell \in \mathcal{H}_\ell$. To come back to our 2 levels example, $\mathbf{x}$ would be $(x_0, x_1) = (a, d)$. We assume that g is a separable function with respect to this direct sum, so that for all $\mathbf{x} \in \mathcal{H}$,

$$g(\mathbf{x}) = \sum_{\ell=1}^{L} g_\ell(x_\ell), \tag{7.16}$$

where for all ℓ, $g_\ell : \mathcal{H}_\ell \to (-\infty, +\infty]$, and its proximity operator is available. This block separability has been exploited to obtain fast solvers for Problem (7.15). These methods have been shown on numerous occasions to be competitive on structured optimization, for instance on problems of the following form:

$$\arg\min_{\mathbf{x}} f(\mathrm{A}\mathbf{x}) + \sum_{\ell=1}^{L} g_\ell(x_\ell), \tag{7.17}$$

where the structure of the matrix A can be exploited to compute global gradient steps at the block scale (thus reducing greatly the computation cost of one update) [191–194]. Consider for instance a blurring matrix A, computing the gradient with respect to one coordinate of x only requires the knowledge of the neighboring coordinates.

Likewise, in our motivating example in section 7.2, if we can pre-compute for instance $\Pi_V \mathrm{A}^* \mathrm{A} \Pi_V^*$, which is of smaller size than $\mathrm{A}^*\mathrm{A}$, then the computation of a block gradient step is lower than the computation of a full gradient step.

The idea of splitting the optimization problem into smaller operations is ubiquitous in practice and has sparked in the last years a lot of research to better understand its potential from a theoretical perspective. The following paragraphs describe the bulk of these studies in the context of block-coordinate forward-backward algorithm where block

updates are done using proximal-gradient descent. A more complete overview of the update methods may be found in [192]. Methods are classified according to their update rules, convergence guarantees, and strategies followed to prove the convergence.

7.3.1 Block-coordinate forward-backward algorithm

The most general formulation of a block-coordinate forward-backward algorithm is the following. Let $(\varepsilon^n)_{n\in\mathbb{N}} = (\varepsilon_1^n, \dots, \varepsilon_L^n)_{n\in\mathbb{N}}$ be a sequence of variables with value in $\{0,1\}^L$. Let $(\tau_\ell)_{1\leq\ell\leq L} \in \mathbb{R}_{++}^L$ and $\mathbf{x}^0 = (x_1^0, \dots, x_L^0) \in \operatorname{dom} g$. Iterate

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \quad \text{for } \ell = 1, \dots, L \\ \quad\quad x_\ell^{n+1} = x_\ell^n + \varepsilon_\ell^n \left(\operatorname{prox}_{\tau_\ell g_\ell} \left(x_\ell^n - \tau_\ell \nabla_\ell f(\mathbf{x}^n) \right) - x_\ell^n \right). \end{array} \tag{7.18}$$

We now review the types of update rules that have been studied in the literature.

Update rules in Algorithm (7.18). This algorithm can either be

- stochastic by choosing randomly $(\varepsilon_1^n, \dots, \varepsilon_L^n) \in \{0,1\}^L$, thus enabling random parallel updates, for all $n \in \mathbb{N}$;
- essentially[2] cyclic by ensuring that for all n only one $\ell \in \{1, \dots, L\}$ is such that $\varepsilon_\ell^n = 1$ (these updates encompass alternated optimization techniques) and in a way that in a cycle all the blocks have been updated at least once;
- parallel and essentially cyclic by setting *a priori* the sequence $(\varepsilon_1^n, \dots, \varepsilon_L^n)$ for all $n \in \mathbb{N}$. In this case for a given n multiple $\ell \in \{1, \dots, L\}$ can be such that $\varepsilon_\ell^n = 1$.

A random shuffling of the order of the updates is also possible at the beginning of each cycle for the last two methods. The convergence guarantees vary depending on the type of updates.

7.3.2 Convergence studies

There have been numerous works to study Algorithm (7.18) in the first two settings. We list some of them below before discussing in depth the convergence guarantees they provide:

- stochastic: [192, 194–203]
- essentially cyclic with/without random shuffling: [59, 64, 80, 179, 191–193, 204–208]

[2]Essentially refers to the fact that what matters for convergence of the algorithm is the existence of a finite number of iterations after which all blocks have been updated once. This number of iterations may be different from the number of blocks. In contrast the adjective cyclic only refers to the sequential update of one block after the other until every one of them has been updated once.

This list of references is not exhaustive, but it is representative of the proof techniques used to study the convergence of Algorithm (7.18). The last setting, to the best of our knowledge, i.e., parallel and cyclic/essentially cyclic with or without random shuffle, has not been studied previously in the literature, and we will consider it in the following. We present now the proof techniques and convergence results used in the first two settings: stochastic and cyclic/essentially cyclic.

Stochastic setting. For most of the literature on randomized approaches only convergence or rate of convergence, in expectation, of the objective function values is shown [192, 197–199].

In order to show convergence of the iterates to a minimizer, the concept of stochastic Quasi-Féjer sequence was introduced in [196]. This framework is powerful and can be applied to many types of block-coordinate descent algorithms (see for instance primal-dual ones in [195]). However, it is thus far only applicable if Ψ is convex, and we will consider the non-convex setting in the following.

It is interesting to remark that in the convex case randomized block-coordinate descent algorithm have shown themselves to be easier to study than their cyclic counterparts, whether it is in terms of complexity, convergence of objective function values, or convergence of the iterates [191, 192, 196, 197, 209]. Even on problems that are solvable by both approaches, practical performance is equivalent [209] or even better for clever selection rules in the cyclic case (Gauss-Southwell rule[3] for instance [193][4] or random reshuffling at each cycle [202][5]), while convergence guarantees are not on par.

Remark 14. *It seems accepted in the literature that ensuring convergence of a random block-coordinate descent algorithm requires proper sampling [194, 198, 210, 211]. Proper sampling refers to the fact that the probability of selecting a block should depend on the Lipschitz constant of the gradient of the block.*

Cyclic setting. Guarantees of convergence of cyclic/essentially cyclic BCD have been investigated for instance in [64, 80, 208] in the framework developed around Kurdyka-Łojasiewicz (KŁ) or Łojasiewicz properties/inequalities [64, 212–214]. The authors proved the convergence to a minimizer or a critical point of Ψ, and some rate of convergence of the sequence of iterates if parameters governing KŁ inequalities are known. The possibility of parallel updates was not investigated in this context even though the significant appeal of block-coordinate descent methods is partly rooted in parallelization.

What we want to do. Guided by the possible analogy between multilevel algorithm and BCD algorithms, we aim to design a convergent block-coordinate descent algorithm

[3] Gauss-Southwell rule: greedy selection of the coordinates to update, i.e., coordinates with the largest gradient norms.

[4] In [193], the authors argue that the Gauss-Southwell selection rule for coordinate descent algorithm tends to perform substantially better than random selection. They identify the subclass of functions that was hindering the potential performance of the Gauss-Southwell rule, that was making it theoretically as effective as random selection, and thus worse in practice due to its cost.

[5] Random permutations in cyclic coordinate descent allows the latter to match performance of random coordinate descent [202]. For quadratic function it was shown before that a significant gap existed between cyclic coordinate descent (without permutations) and random coordinate descent [203].

for non-smooth and non-convex optimization where the updates are potentially parallel and may be randomly shuffled if needed (as it in can be useful in practice). This will allow us to implement update strategies mimicking those of multilevel algorithms.

Hence, we propose an unbalanced (i.e., some blocks are updated more often than others) parallel block-coordinate forward-backward algorithm, which we refer to as Hierarchical Block-Coordinate Forward-Backward (H-BC-FB) in reference to its underlying hierarchical block selection rule.

The first major contribution is to show the convergence of the method both in function values and with respect to the set of critical points, even when Ψ is non-convex. The convergence of the proposed scheme is analyzed from two point of views: we propose a KŁ-based analysis in the general setting and stochastic convergence guarantees in the convex case. Both analyzes allow us to provide convergence of the objective function values, and of the iterates to a critical point of Ψ. Even though the deterministic setting can be seen as a particular instance of the stochastic one, the study of the sequences generated by a stochastic block algorithm rely on specific tools that have limitations. As said before the concept of stochastic Quasi-Féjer sequence [196, Proposition 2.3] does not work yet in the non-convex setting. Moreover, the independence between consecutive iterations of the random variables selecting the blocks is crucial to establish convergence properties (see [194, 195]). As we aim to allow correlations in the block selection between iterations (remember our example in Section 7.2), so that the classical stochastic tools are not applicable here (at least not in a straightforward way).

The second contribution amounts to show that the proposed hierarchical block architecture encompasses the classical multilevel formalism. This new point of view, provided by the block-coordinate descent framework, allows us to define key points of the multilevel algorithm in a rigorous manner, and most importantly *a priori*. This will be presented in Section 7.5.

7.3.3 Relevant notations and technical background

In this section we introduce the necessary mathematical background and notations that will be used We denote by $|||\cdot|||$ the Euclidean norm on $\mathcal{H}$ and by $\|\cdot\|$ the Euclidean norm on the L spaces $(\mathcal{H}_\ell)_{1\leq\ell\leq L}$. Similarly, the scalar product on $\mathcal{H}$ will be denoted by $\langle\langle\cdot,\cdot\rangle\rangle$ and the scalar product on $\mathcal{H}_\ell$ by $\langle\cdot,\cdot\rangle$; the potential ambiguity between two spaces is cleared up as the variables on which the scalar product is applied will be indexed by ℓ. Note that for all $\mathbf{x},\mathbf{y}\in\mathcal{H}$, $\langle\langle\mathbf{x},\mathbf{y}\rangle\rangle = \sum_{\ell=1}^{L}\langle x_\ell, y_\ell\rangle$. For a continuously differentiable function f, we denote for all $\mathbf{x}\in\mathcal{H}$ by $\nabla_\ell f(\mathbf{x})$ the gradient of f taken with respect to the variables in the ℓ-th block.

We will index our sequence of iterates by a superscript denoting the iteration number and a subscript denoting the block of variables. Thus, x_ℓ^n denotes the ℓ-th block at the n-th iteration. For convenience, we will write $\mathbf{x}^n$ to denote the full variable at iteration n, so that $\mathbf{x}^n = (x_1^n,\dots,x_L^n)$. Furthermore, the convergence analysis relies on a cyclic rule for the updates, and we assume that each cycle consists of K maximum iterations. We will denote with exponent k, with an upper bar and in bold font the iterates $\bar{\mathbf{x}}^k$ that has seen k cycles and thus $k\times K$ iterations to accentuate the difference with the iterates $\mathbf{x}^n$, whose sequence is not guaranteed to converge. Thus, $\bar{\mathbf{x}}^k = \mathbf{x}^n$ when $n = k\times K$. These notations are summarized in Table 7.1 and are illustrated in Figure 7.3.

x_ℓ^n	block in $\mathcal{H}_\ell$ at iteration n
$\mathbf{x}^n$	iterate in $\mathcal{H}$ at iteration n
$\bar{\mathbf{x}}^k$	iterate in $\mathcal{H}$ at cycle number k

Table 7.1: Summary of notations of the iterates generated by our algorithm

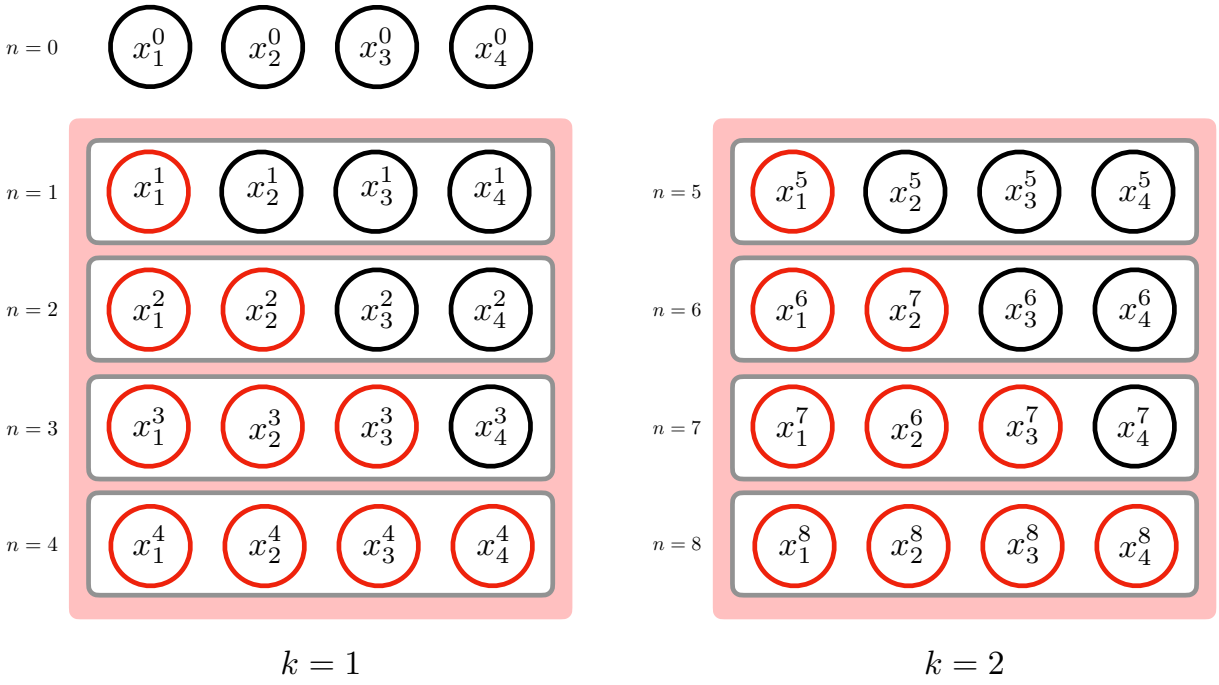

Figure 7.3: One possibility of a complete iteration of the proposed Hierarchical Block-Coordinate Forward-Backward for a variable x separable into 4 blocks. First row is the initialization of the algorithm. Each row indicates an iteration n. The group of blocks that are updated is highlighted with a red circle. $K = 4$ iterations are required to complete the first cycle $k = 1$ which is highlighted in rose. Each cycle is a multilevel cycle, where x_1 would be the coarsest block while x_4 is the finest. The union of x_1, x_2, x_3, and x_4 would constitute our fine level.

As we venture in the non-convex setting, we will need an appropriate notion of sub-gradient.

Definition 32. ***Subgradient [52].*** *Let $g : \mathcal{H} \mapsto \mathbb{R}$, and let $\mathbf{x} \in \mathcal{H}$. The Fréchet subdifferential of g at $\mathbf{x}$ is denoted by $\hat{\partial} g(\mathbf{x})$ and is given by*

$$\hat{\partial} g(\mathbf{x}) = \left\{ \hat{v}(\mathbf{x}) \in \mathcal{H} | \lim_{\mathbf{y} \to \mathbf{x}} \inf_{\mathbf{y} \neq \mathbf{x}} \frac{1}{|||\mathbf{x} - \mathbf{y}|||} \left(g(\mathbf{y}) - g(\mathbf{x}) - \langle \mathbf{y} - \mathbf{x}, \hat{v}(\mathbf{x}) \rangle \right) \geq 0 \right\} \tag{7.19}$$

If $\mathbf{x} \notin$ dom g, then $\hat{\partial} g(\mathbf{x}) = \emptyset$. The limiting subdifferential of g at $\mathbf{x}$ is denoted by $\partial g(\mathbf{x})$ and is given by

$$\partial g(\mathbf{x}) = \Big\{ v(\mathbf{x}) \in \mathcal{H} | \exists \left(\mathbf{x}^k, \hat{v}(\mathbf{x}^k) \right) \to (\mathbf{x}, v(\mathbf{x})) \tag{7.20}$$

$$\text{such that } g(\mathbf{x}^k) \to g(\mathbf{x}) \text{ and } (\forall k \in \mathbb{N})\ \hat{v}(\mathbf{x}^k) \in \hat{\partial} g(\mathbf{x}^k) \Big\}. \tag{7.21}$$

Recall that if g is convex, its subdifferential is given for all $\mathbf{x} \in \mathcal{H}$ by

$$\partial g(\mathbf{x}) = \{ \mathbf{s} \in \mathcal{H}, g(\mathbf{x}) + \langle \mathbf{s}, \mathbf{y} - \mathbf{x} \rangle \leq g(\mathbf{y}), \forall \mathbf{y} \in \mathcal{H} \}. \tag{7.22}$$

Both $\hat{\partial} g(x)$ and $\partial g(x)$ are closed [52, Theorem 8.6].

The Kurdyka-Łojasiewicz (KŁ) property. We remind here the definition of the KŁ property and some other important properties. First, we introduce the notion of sublevel sets.

Definition 33. ***Sublevel sets [64].*** *Given $a, b \in \mathbb{R}$ and Ψ a proper lower semicontinuous function, we set*

$$[a \leq \Psi \leq b] := \{ x \in \mathbb{R}^N,\ a \leq \Psi(x) \leq b \}$$

Concave and continuous functions of the following form are of particular interest in the KŁ framework: they are called the desingularizing functions.

Definition 34. ***Concave and continuous functions [64].*** *Let $\eta \in (0, +\infty]$. We denote by Φ_η the class of all concave and continuous functions $\varphi : [0, \eta) \to \mathbb{R}_+$ which satisfy the following conditions*

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

(ii) φ *is* C^1 *on* $(0, \eta)$ *and continuous at* 0,

(iii) for all $s \in (0, \eta)$: $\varphi'(s) > 0$

Now, we can introduce the definition of a KŁ function.

Definition 35. ***Kurdyka-Łojasiewicz (KŁ) property [64].*** *Let $\Psi : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous.*

(i) The function Ψ is said to have the KŁ property at $\bar{u} \in$ dom $\partial\Psi$ if there exist $\eta \in (0, +\infty]$, a neighborhood U of $\bar{u}$ and a function $\varphi \in \Phi_\eta$ such that for all

$$u \in U \cap [\Psi(\bar{u}) < \Psi(u) < \Psi(\bar{u}) + \eta],$$

the following inequality holds

$$\varphi'(\Psi(u) - \Psi(\bar{u}))\, dist(0, \partial\Psi(u)) \geq 1 \tag{7.23}$$

(ii) If Ψ satisfies the KŁ property at each point of dom $\partial\Psi$ then Ψ is called a KŁ function.

The following lemma characterizes the KŁ property in a more practical form.

Lemma 19. ***Uniformized KŁ property [64].*** *Let Ω be a compact subset of $\mathbb{R}^N$. Let $\Psi : \mathbb{R}^N \mapsto (-\infty, +\infty]$ be a proper and lower semicontinuous function, constant on Ω and satisfying the KŁ inequality on Ω. Then there exists $\varepsilon > 0, \eta > 0$ and $\varphi \in \Phi_\eta$ such that for all $\bar{u} \in \Omega$ and all u in the following intersection*

$$\left\{u \in \mathbb{R}^N : dist(u, \Omega) < \varepsilon\right\} \cap [\Psi(\bar{u}) < \Psi(u) < \Psi(\bar{u}) + \eta] \tag{7.24}$$

one has,

$$\varphi'(\Psi(u) - \Psi(\bar{u}))\, dist(0, \partial\Psi(u)) \geq 1. \tag{7.25}$$

Remark 15. *The KŁ property is satisfied by numerous classes of functions, and notably for those considered in typical optimization settings such as ℓ_p norms. See [214] for an overview on this property.*

Remark 16. *The KŁ property may seem obscure at a first glance, but in essence, it says that a function respecting this property should not be too flat around its critical point [64, 214]. We attempt a hand-wavy explanation here. For a rigorous and far more complete point of view see for instance [61, Chapter 3].*

The flatness of a function around its critical points is characterized by the norm of its gradient. In a continuous setting, one can analyze typical descent algorithm such as gradient descent on continuously differentiable function Ψ using the following differential equation:

$$(\forall t \geq 0) \quad \dot{x}(t) + \nabla\Psi(x(t)) = 0. \tag{7.26}$$

This equation can be obtained by seeing gradient descent as an Euler forward scheme to compute a discretized solution of an ordinary differential equation (ODE):

$$x_{t+1} = x_t - \tau\nabla\Psi(x_t) \tag{7.27}$$

$$\frac{x_{t+1} - x_t}{\tau} = -\nabla\Psi(x_t). \tag{7.28}$$

By taking the limit when τ goes to 0 in Equation (7.28), we recover (7.26).

Briefly, $t \mapsto \|\nabla\Psi(x(t))\|$ *will control the length of the trajectory*

$$\int_0^{+\infty} \|\dot{x}(t)\|dt,$$

through the relationship of Equation (7.26). The function φ *in Definition 34 tells us how* $\|\nabla\Psi(x(t))\|$ *behaves if* Ψ *is such that the KŁ property is respected through Equality (7.25) (*$dist(0, \partial\Psi(x)) = \|\nabla\Psi(x)\|$*). Therefore,*

$$\int_0^{+\infty} \|\dot{x}(t)\|dt < +\infty.$$

This means that the sequence has finite length and thus converges to a critical point [61, Chapter 3]. This analysis is generalizable to non-smooth functions using the subdifferential instead of the gradient in the differential equation [61].

7.4 Convergence of the Hierarchical-BC-FB algorithm

In this section, we study Algorithm (7.18) in the "deterministic" setting. The proof of convergence relies on the framework developed in [64, 213, 215], which is built on the KŁ property [64, 213, 216]. More precisely, our convergence proof follows the same structure as the proof presented in [64] for an alternated proximal minimization algorithm.

Therefore, for clarity, we defer the technical details of the proof to the Appendix A.4 and focus on the main ideas in the following.

7.4.1 Convergence settings

Assumptions on the functions. The convergence relies on several classical assumptions that we present in the following.

Assumption 4.

A1 $\Psi := f + \sum_{i=1}^{L} g_\ell$ *is coercive, and bounded below. For all* ℓ, g_ℓ *is bounded below, as well as* f.

A2 Ψ *satisfy the KŁ property (Definition 35 and Lemma 19).*

Assumption 5.

A3 *For all* ℓ, g_ℓ *is a lower semicontinuous, proper function. Its proximity operator is available under closed form.*

A4 f *is continuously differentiable and there exists* $(\beta_{\ell,j})_{\ell,j\in\{1,\dots,L\}} \in \mathbb{R}_{++}$ *such that*

$$(\forall \ell, j \in \{1,\dots,L\})(\forall \mathbf{x} \in \mathcal{H})(\forall v_j \in \mathcal{H}_j) \\ \|\nabla_\ell f(\mathbf{x} + (0,\dots,0,v_j,0,\dots,0)) - \nabla_\ell f(\mathbf{x})\| \leq \beta_{\ell,j}\|v_j\| \quad (7.29)$$

Remark 17. *With Assumption* A3*, we restrict ourselves to the case where the proximity operators of* g_ℓ *is available explicitly for all* ℓ*, but may be set-valued without convexity assumptions (see Appendix A.4, Lemma 35). It is quite standard in a non-convex setting, but remains a more restrictive assumption than the one made to define IML FISTA (see Chapter 4).*

Remark 18. *Assumption* A1 *is sufficient to assert that the sequences generated by our algorithm are bounded [213]. Assumption* A4 *is fairly easy to verify in practice, as it is implied by having ∇f being Lipschitz continuous: if ∇f is Lipschitz continuous with constant β_f, then we can take $\beta_{\ell,j} = \beta_f$ for all ℓ, j.*

Assumption A4 states that every partial gradient with respect to the block is Lipschitz continuous with respect to all the blocks, which is a quite stronger assumption than being Lipschitz continuous with respect only to its block. From this assumption we can derive what we call multiple block smoothness, a common assumption in the BCD literature (e.g. [194, Assumption S1-S2-S3]).

Proposition 6. ***Multiple block smoothness.*** *Suppose that Assumption 5 holds. For all $\varepsilon = (\varepsilon_\ell)_{1\leq\ell\leq L} \in \{0,1\}^L$, there exists $\beta > 0$ such that for all $1 \leq \ell \leq L$, $v_\ell \in \mathcal{H}_\ell$ we have*

$$(\forall \mathbf{x} \in \mathcal{H}) \quad \left\|\nabla f(\mathbf{x} + (\varepsilon_\ell v_\ell)_{\ell\in\{1,\dots,L\}}) - \nabla f(\mathbf{x})\right\| \leq \beta \left\|(\varepsilon_\ell v_\ell)_{\ell\in\{1,\dots,L\}}\right\| \tag{7.30}$$

Assumptions on the update rules. We consider an essentially cyclic update scheme for the blocks in which parallel updates of different blocks may be used, paired with a potential shuffle of the updates order, as specified in the following assumption.

Assumption 6. *Consider the following assumptions on the update rules:*

A5 *Every K iterations, each block has been updated at least once, i.e., updates are essentially cyclic. Formally, denote by I^n the set of the blocks updated at this iteration: $I^n = \{\ell \mid \varepsilon_\ell^n = 1\} \subseteq \{1,\dots,L\}$. For all j*

$$\bigcup_{n=j}^{j+K-1} I^n = \{1,\dots,L\} \tag{7.31}$$

Remark 19. *Assumption* A5 *on the order of the updates of the blocks is not restrictive. For instance, it includes the sequential update of the blocks if $K = L$ and the classical forward-backward update for $K = 1$.*

Now, to allow flexibility, one may want to alternate between different ways of updating the blocks from one cycle to the other. For instance, from a multilevel setting, alternating between V-cycles (Figure 7.3) and W-cycles may be interesting [85].

Furthermore, it has been noted that randomly shuffling the order of updates could improve greatly the convergence speed of the algorithm in practice [208]. Such shuffle is compatible with Assumption A5. Our algorithm is thus as flexible as possible in the non-randomized setting.

Our algorithm. For the purpose of clarity, we rewrite algorithm (7.18) to incorporate explicitly the cycles.

Let $K \in \mathbb{N}^*$ be the number of iterations to complete one cycle. Let $(\varepsilon^n)_{n\in\mathbb{N}} = (\varepsilon_1^n,\dots,\varepsilon_L^n)_{n\in\mathbb{N}}$ be a sequence of variables with value in $\{0,1\}^L$. Let $(\tau_\ell)_{1\leq\ell\leq L} \in \mathbb{R}_{++}^L$

and $\bar{\mathbf{x}}^0 = \mathbf{x}^0 = (x_1^0, \dots, x_L^0) \in \operatorname{dom} g$. Set $k = 0$. Iterate

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \quad \text{for } \ell = 1, \dots, L \\ \quad\quad x_\ell^{n+1} = x_\ell^n + \varepsilon_\ell^n \left(\operatorname{prox}_{\tau_\ell g_\ell} \left(x_\ell^n - \tau_\ell \nabla_\ell f(\mathbf{x}^n) \right) - x_\ell^n \right). \\ \quad \text{if } n + 1 \equiv 0 \ [K] \\ \quad\quad k = k + 1 \\ \quad\quad \bar{\mathbf{x}}^k = \mathbf{x}^{n+1} \end{array} \tag{7.32}$$

Convergence proof. Before setting the main result, we need two intermediary lemmas that show that each iteration of algorithm H-BC-FB decreases the objective function Ψ, and that the set of limit points of the iterates produced by the algorithm is contained in the set crit Ψ of the critical points of Ψ, i.e., the set of points $\mathbf{x}$ such that $0 \in \partial\Psi(\mathbf{x})$.

The following lemma bounds the norm of the difference between *cycle* iterates by the sum of all the norm of the differences between *block* iterates.

Lemma 20. *Let $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm H-BC-FB. Then,*

$$\left\| \bar{\mathbf{x}}^{k+1} - \bar{\mathbf{x}}^k \right\| \leq \left(\sum_{n=k\times K}^{(k+1)\times K-1} \sum_{\ell \in I^n} \| x_\ell^{n+1} - x_\ell^n \| \right) \tag{7.33}$$

Proof. Proof is in Appendix A.4. □

The next lemma shows that the function value decreases at each cycle iteration.

Lemma 21. ***Sufficient decrease property.*** *Suppose that Assumption 5 holds. Let $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ be the sequence of cycle iterates generated by algorithm H-BC-FB. Let $n \in \mathbb{N}$, $\mathbf{x}^n = (x_1^n, \dots, x_L^n)$ the n-th iterate of algorithm H-BC-FB (7.18).*
For each n, let $\beta_f^n := \sqrt{\sum_{j\in J_n, 1\leq\ell\leq L} \varepsilon_j \beta_{\ell,j}^2}$ and $0 < \tau_\ell^n < 1/\beta_f^n$. Then

$$\Psi(\bar{\mathbf{x}}^{k+1}) + \left(\sum_{n=k\times K}^{(k+1)\times K-1} \sum_{\ell \in I^n} \frac{1}{2} \left(\frac{1}{\tau_\ell^n} - \beta_f^n \right) \| x_\ell^n - x_\ell^{n+1} \|^2 \right) \leq \Psi(\bar{\mathbf{x}}^k). \tag{7.34}$$

Furthermore,

$$\sum_{n=0}^{+\infty} \left(\sum_{\ell=1}^{L} \| x_\ell^n - x_\ell^{n+1} \|^2 \right) < +\infty, \tag{7.35}$$

which implies
$\lim_{n\to+\infty} \| x_\ell^n - x_\ell^{n+1} \| = 0$ *for all* ℓ *and thus* $\lim_{n\to+\infty} \| \mathbf{x}^n - \mathbf{x}^{n+1} \| = 0$.

Proof. Proof is in Appendix A.4. □

Remark 20. *Note that a random shuffle of the update order at each cycle is possible, as the order of the updates does not intervene in the proof of sufficient decrease nor in the*

proof of the following results. A similar observation was made in [208] (but without the possibility of parallel updates). Also note that a random shuffle is totally different from a random choice of block update at each iteration, as in the latter case there is no definite guarantee that a cycle of length K where all blocks have been updated, would exist (only with high probability).

Remark 21. *An interesting consequence of this lemma is the fact that it ensures that* ***multilevel steps****, for our motivating example (Section 7.2),* ***are always decreasing the function value****. This was guaranteed only up to an error term for IML FISTA (Chapter 4, Lemma 10).*

7.4.2 Main result

Now that we have established the decrease of the objective function at each iteration, we are ready to state our main result.

Theorem 8. ***Sufficient decrease and subgradient bound.*** *Suppose that Assumptions 4 and 5 hold. Let $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm H-BC-FB. The following assertions hold.*

(i) The sequence $\{\Psi(\bar{\mathbf{x}}^k)\}_{k\in\mathbb{N}}$ is non-increasing. For each n, let $\beta_f^n = \sqrt{\sum_{j\in J_n, 1\leq\ell\leq L}\varepsilon_j\beta_{\ell,j}^2}$ and $0<\tau_\ell^n<1/\beta_f^n$. Then for all $k\geq 0$

$$\Psi(\bar{\mathbf{x}}^{k+1}) + \left(\sum_{n=k\times K}^{(k+1)\times K-1}\sum_{\ell\in I^n}\frac{1}{2}\left(\frac{1}{\tau_\ell^n}-\beta_f^n\right)\|x_\ell^n - x_\ell^{n+1}\|^2\right) \leq \Psi(\bar{\mathbf{x}}^k). \tag{7.36}$$

(ii) For each $k\in\mathbb{N}$ define

$$\bar{B}^{k+1} = \left(\frac{x_{\ell,n_\ell} - x_{\ell,n_{\ell+1}}}{\tau_{\ell,n_\ell}} - \nabla_\ell f(x_{n_\ell}) + \nabla_\ell f(\bar{\mathbf{x}}^{k+1})\right)_{1\leq\ell\leq L} \tag{7.37}$$

where n_ℓ is a positive integer such that $k\times K\leq n_\ell\leq(k+1)\times K-1$, and is the last iteration of cycle k at which block ℓ will receive an update. Then $\bar{B}^{k+1}\in\partial\Psi(\bar{\mathbf{x}}^{k+1})$ and there exist positive numbers τ_k such that:

$$\left\|\bar{B}^{k+1}\right\| \leq \left(\frac{1}{\tau_k}+\beta_f\right)\left(\sum_{n=k\times K}^{(k+1)\times K-1}\sum_{\ell\in I^n}\frac{1}{2}\left(\frac{1}{\tau_\ell^n}-\beta_f^n\right)\|x_\ell^n - x_\ell^{n+1}\|\right). \tag{7.38}$$

Proof. Proof is in Appendix A.4. □

Theorem 9. ***Converge of the sequence to critical points of*** Ψ**.** *Suppose that Assumptions 4 and 5 hold. Let* $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ *be a sequence generated by Algorithm H-BC-FB. The following assertions hold.*

(iii) The sequence $(\bar{\mathbf{x}}^k)$ *has finite length, that is,*

$$\sum_{k=1}^{\infty} \left\| \bar{\mathbf{x}}^k - \bar{\mathbf{x}}^{k+1} \right\| < \infty. \tag{7.39}$$

(iv) The sequence $(\bar{\mathbf{x}}^k)$ *converges to a critical point* $\mathbf{x}^*$ *of* Ψ*.*

Proof. Proof is in Appendix A.4. □

The proof of the convergence of the sequence require the study of the limit points set, defined as follows.

Definition 36. ***Limit points set [64].*** *The set of all limit points of sequences generated by H-BC-FB from a starting point* $\mathbf{x}^0 = \bar{\mathbf{x}}^0$ *will be denoted by* $\mathrm{lp}(\bar{\mathbf{x}}^0)$*:*

$$\mathrm{lp}(\bar{\mathbf{x}}^0) = \{\mathbf{x}^* \in \mathcal{H}, \exists \text{ an increasing sequence of integers } \{k_j\}_{j\in\mathbb{N}}, \text{ such that } \bar{\mathbf{x}}^{k_j} \to \mathbf{x}^* \text{ as } j \to +\infty\}$$

The properties of the limit points of sequences produced by block algorithms such as the proposed H-BC-FB were investigated in [64], small tweaks are required here.

Lemma 22. ***Properties of the limit points set.*** *Suppose that Assumptions 4 and 5 hold. Let* $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ *be a sequence generated by Algorithm H-BC-FB starting from* $\bar{\mathbf{x}}^0 = \mathbf{x}^0$*. The following hold:*

(i) $\emptyset \neq \mathrm{lp}(\bar{\mathbf{x}}^0) \subset$ *crit* Ψ

(ii) We have

$$\lim_{k\to\infty} \mathrm{dist}(\bar{\mathbf{x}}^k, \mathrm{lp}(\bar{\mathbf{x}}^0)) = 0. \tag{7.40}$$

(iii) $\mathrm{lp}(\bar{\mathbf{x}}^0)$ *is a nonempty, compact and connected set.*

(iv) The objective function Ψ *is finite and constant on* $\mathrm{lp}(\bar{\mathbf{x}}^0)$*.*

Proof. Proof is in Appendix A.4. □

Sketch of the proof for our main result. We split the proof of our main result into 4 steps, which are common when studying descent algorithm on KŁ functions [64, 179, 208, 212, 213]. For each step we detail how the existing proofs were adapted for our approach.

(i) **Sufficient decrease property:** we will show that at each cycle, the objective function Ψ is decreased. The decrease is controlled by the squared norm of the differences between the block updates. *Difference with the literature:* we introduce

the possibility of parallel updates to decrease the function, and thus to adapt the choice of step size to the smoothness of the group of blocks considered.

(ii) **Subgradient upper bound:** at each cycle, we can exhibit an upper bound on one element of the subgradient of Ψ at the cycle iterate. This upper bound is controlled by the norm of the differences between the block updates. *Difference with the literature:* this bound is not sharp at all (as the reader can see it in the proof in Appendix A.4), but it is necessary to write it in this way to apply the KŁ property and obtain finite length.

(iii) **Limit points are critical points:** the set of limit points of the sequences generated by our algorithm will be a subset of the set of critical points of Ψ. *Difference with the literature:* due to the possible parallel block updates, we need to be a bit more cautious when looking at the converging subsequences.

(iv) **Finite length of the sequences:** the sequences generated by our algorithm have finite length and thus converge. This is a consequence of the KŁ property satisfied by Ψ (see Definition 35) and of point (i) and (ii). *Difference with the literature:* we invoke a particular instance of Cauchy-Schwartz inequality to obtain the desired result.

Remark 22. *Regarding Theorems 8 and 9, we can make the following remarks:*

- *The KŁ property allows deriving the convergence rate of the iterates to a critical point of Ψ when for instance f and g_ℓ, for all ℓ, are semi-algebraic [212]. The class of semi-algebraic function is quite large [64, 212, 215]. The desingularizing function φ can be defined with parameters that control this convergence rate (see [64, Remark 6]). Finding φ is a non-trivial question, and papers are dedicated to its computation for classical classes of function, see for instance [217].*

- *It may not appear obvious that algorithm H-BC-FB needs to update each block in an essentially cyclic manner, but this is necessary otherwise point (ii) of Theorem 8 would not hold: if a block is never updated the norm of $A^k \in \partial\Psi(\bar{\mathbf{x}}^k)$ will not go to 0 as k goes to infinity.*

Convexity of the regularization. We can derive a slightly different sufficient decrease property of our algorithm when assuming convexity of the regularizing functions g_ℓ, for all ℓ. This assumption allows us to take bigger step sizes when updating the blocks.

Lemma 23. ***Sufficient decrease property: convexity of*** g_ℓ**.** *Suppose that Assumption 5 holds. Suppose also that for all* $\ell \in \{1, \dots, L\}$, g_ℓ *is a convex function. Let* $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ *be the sequence of cycle iterates generated by algorithm H-BC-FB. Let* $n \in \mathbb{N}$, $\mathbf{x}^n = (x_1^n, \dots, x_L^n)$ *the n-th iterate of algorithm H-BC-FB (7.18). For each n, let* $\beta_f^n = \max_{\ell\in I^n} \nu_\ell$ *and* $0 < \tau_\ell^n < 2/\beta_f^n$*. Then*

$$\Psi(\bar{\mathbf{x}}^{k+1}) + \left(\sum_{n=k\times K}^{(k+1)\times K-1} \sum_{\ell\in I^n} \left(\frac{1}{\tau_\ell^n} - \frac{\beta_f^n}{2} \right) \|x_\ell^n - x_\ell^{n+1}\|^2 \right) \leq \Psi(\bar{\mathbf{x}}^k). \tag{7.41}$$

Furthermore,

$$\sum_{n=0}^{+\infty} \left(\sum_{\ell=1}^{L} \|x_\ell^{n+1} - x_\ell^n\|^2 \right) < +\infty \tag{7.42}$$

which implies $\lim_{n\to+\infty} \|x_\ell^n - x_\ell^{n+1}\| = 0$ *for all* ℓ *and thus* $\lim_{n\to+\infty} |||\mathbf{x}^n - \mathbf{x}^{n+1}||| = 0$.

Proof. Proof is in Appendix A.4. □

We have seen how to construct a convergent hierarchical block-coordinate forward-backward algorithm, able to handle non-convexity and non-smoothness of the objective function. This algorithm is deterministic by essence, even though a random shuffle of the order of the updates is possible. Most of the literature on block-coordinate descent algorithms is more concerned by stochastic algorithms, as they provided more guarantees under more general update rules (e.g. parallel updates). For our presentation of the BC point of view of multilevel algorithms to be complete, we need to inspect multilevel algorithm from the stochastic perspective. This is the subject of the next section.

7.4.3 Convergence of H-BC-FB in a stochastic setting

In this section, we briefly present a convergence result for a randomized version of our Hierarchical Block-Coordinate Forward-Backward algorithm. The convergence result in itself is a direct application of [194, Theorem 4.9]. We aim here to construct a stochastic BC FB that, in expectation, mirrors the behavior of our multilevel algorithm and is convergent. Such algorithm follows classic rules of stochastic BCD algorithms that can update in parallel the blocks.

With such algorithm we will be able to have a complete comparison of the update rules available today for BC descent algorithms. Recall that the algorithm is of the following form: Let $(\varepsilon^n)_{n\in\mathbb{N}} = (\varepsilon_1^n, \dots, \varepsilon_L^n)_{n\in\mathbb{N}}$ be a sequence of variables with value in $\{0,1\}^L$. Let $(\tau_\ell)_{1\leq\ell\leq L} \in \mathbb{R}_{++}^L$ and $\mathbf{x}^0 = (x_1^0, \dots, x_L^0) \in \operatorname{dom} g$. Iterate

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \quad \text{for } \ell = 1, \dots, L \\ \qquad x_\ell^{n+1} = x_\ell^n + \varepsilon_\ell^n \left(\operatorname{prox}_{\tau_\ell g_\ell} \left(x_\ell^n - \tau_\ell \nabla_\ell f(\mathbf{x}^n) \right) - x_\ell^n \right). \end{array} \tag{7.43}$$

With a main difference: ε and $\mathbf{x}$ are now random variables. Consider the following assumptions:

Assumption 7.

A6 $f : \mathcal{H} \to \mathbb{R}$ *is convex and continuously differentiable,*

A7 *for every* $\ell = 1, \dots, L$, $g_\ell : \mathcal{H}_\ell \to]-\infty, +\infty]$ *is proper, convex, and lower semicontinuous.*

A8 $\varepsilon = (\varepsilon_1, \dots, \varepsilon_L)$ *is a random variable with values in* $\{0,1\}^L$*, such that for every* $\ell \in \{1, \dots, L\}$, $\mathsf{P}(\varepsilon_\ell = 1) > 0$ *and* $\mathsf{P}(\varepsilon = (0, \dots, 0)) = 0$. *Note* $\mathsf{p}_\ell = \mathsf{P}(\varepsilon_\ell = 1)$

We can now present a way to construct update rules to mimic our multilevel algorithm that verify Assumption A8. As we mostly used V-scheme in practice (Chapters 5 and 6), we present an update rule for this scheme.

Lemma 24. ***V-scheme probabilities for H-BC-FB.*** *Suppose that* $\varepsilon = (\varepsilon_1, \dots, \varepsilon_L)$ *is a random variable with values in* $\{0,1\}^L$*, such that for every* $\ell \in \{1, \dots, L-1\}$,

- $\mathsf{P}(\varepsilon_{\ell+1} = 1 | \varepsilon_\ell = 1) > 0$,
- $\mathsf{P}(\varepsilon_{\ell+1} = 1 | \varepsilon_\ell = 0) = 0$,

and that $\mathsf{P}(\varepsilon_1 = 1) = 1$.
Then, for every $\ell \in \{1, \dots, L\}$, $\mathsf{P}(\varepsilon_\ell = 1) > 0$ *and* $\mathsf{P}(\varepsilon = (0, \dots, 0)) = 0$.

Proof. The second point is straightforward. For the first point, simply remark that for every $\ell \in \{2, \dots, L\}$:

$$\mathsf{P}(\varepsilon_\ell = 1) = \mathsf{P}(\varepsilon_\ell = 1 | \varepsilon_{\ell-1} = 1)\mathsf{P}(\varepsilon_{\ell-1} = 1), \tag{7.44}$$

then one directly has:

$$\mathsf{P}(\varepsilon_\ell = 1) = \left(\prod_{j=2}^{j=\ell} \mathsf{P}(\varepsilon_j = 1 | \varepsilon_{j-1} = 1) \right) \mathsf{P}(\varepsilon_1 = 1). \tag{7.45}$$

which is strictly greater than 0. □

One can see that with this construction we will update the coarsest level at each iteration, and that updating "fine" levels will also force us to update coarser levels, which is typical of multilevel methods.

The sampling of ε is done sequentially by increasing ℓ until we reach the first zero occurrence. In order to update all levels as often as possible, the value of $\mathsf{P}(\varepsilon_{\ell+1} = 1 | \varepsilon_\ell = 1)$ should be close to 1 for large ℓ.

Choosing the right value of conditional probabilities. In a typical V-scheme, a multilevel algorithm would compute m iterations at each coarse level, going upwards in the resolution. After that it would compute one iteration at fine level. Thus, we should adjust the conditional probabilities of activating each block so that with high probability we update $m \geq 0$ times the coarsest level alone, then m times the coarsest level and the second to last coarsest level, and so on ... We have for all ℓ:

$$\mathsf{P}(\varepsilon_\ell = 1) = \left(\frac{1}{m} \right)^\ell \tag{7.46}$$

which yields to:

$$\mathsf{P}(\varepsilon_{\ell+1} = 1 | \varepsilon_\ell = 1) = \frac{1}{m} \tag{7.47}$$

Convergence of the stochastic algorithm. We can now state the convergence result for the stochastic version of our algorithm. The proof is a direct application of [194, Theorem 4.9] and is therefore omitted.

Theorem 10. ***Convergence of stochastic and parallel BC FB [194, Theorem 4.9].*** *Let $(\varepsilon_n)_{n\in\mathbb{N}} = (\varepsilon_1^n, \dots, \varepsilon_L^n)_{n\in\mathbb{N}}$ be a sequence of independent copies of ε. Let $(\tau_\ell)_{1\leq\ell\leq L} \in \mathbb{R}_{++}^L$ and $x_0 = (x_{1,0}, \dots, x_{L,0}) \equiv \mathsf{x}_0 \in$ dom g be a constant random variable. Set $\delta = \max_{1\leq\ell\leq L} \tau_\ell \nu_\ell$ and $\mathsf{p}_{min} = \min_{1\leq\ell\leq L} \mathsf{P}(\varepsilon_\ell = 1)$.*
Set $\mathbf{Id} = \bigoplus_{\ell=1}^L \frac{1}{\tau_\ell \mathsf{p}_\ell} \mathrm{Id}_\ell$ (the identity operators on $\mathcal{H}_\ell$), $F_* = \inf F$, and $S_* = \arg\min F \subset \mathcal{H}$. Then the following hold.

(i) $\mathsf{E}[F(\mathbf{x}^n)] \to F_*$.

(ii) Suppose that $S_* \neq \emptyset$. Then $\mathsf{E}[F(\mathbf{x}^n)] - F_* = o(1/n)$ and for every integer $n \geq 1$,

$$\mathsf{E}[F(\mathbf{x}^n)] - F_* \leq \left[\frac{dist_{\mathbf{Id}}^2(x_0, S_*)}{2} + \left(\frac{\max\{1, (2-\delta)^{-1}\}}{\mathsf{p}_{min}} - 1 \right) (F(\mathbf{x}^0) - F_*) \right] \frac{1}{n}$$

Moreover there exists a random variable x_* taking values in S_* such that $\mathbf{x}^n \rightharpoonup x_*$.

7.5 Multilevel algorithms from the BC point of view: the general case

In the rest of the chapter, equipped with the hierarchical block-coordinate descent framework that we developed, we will construct a multilevel algorithm for the wavelet deblurring problem. This algorithm will be compared in its construction to what would be the standard block-coordinate descent algorithms of the literature in order to demonstrate empirically that following the multilevel spirit is indeed a better way of constructing a block-coordinate descent algorithm for problems with similar structures.

By looking at multilevel algorithms through this lens we are able to answer several questions regarding the construction of such algorithms in order to obtain convergence (and thus how to construct them properly). The problem we consider is commonly formulated as:

$$\widehat{\mathbf{x}} \in \operatorname*{Argmin}_{\mathbf{x}\in\mathcal{H}} \Psi(\mathbf{x}) := \frac{1}{2}\|\mathrm{A}\mathbf{x} - z\|_2^2 + \lambda\|\mathrm{D}\mathbf{x}\|_1 \tag{7.48}$$

where D is the wavelet transform and λ is potentially multivalued to penalize scales differently. The structure of the wavelet transform decomposes naturally $\mathbf{x}$ at successive scales that we will employ to define our coarse levels. For clarity of the presentation, we leave the technical proofs to Appendix A.6.

7.5.1 Multiresolution analysis and optimization

In this section we introduce as rigorously as possible, without introducing too much complexity, the wavelet decomposition of an image. This will allow us to present the construction of our multilevel algorithm faithfully.

Let ψ be a wavelet function and consider $\{\psi^i_{j,k}\}$ a Riesz basis where, for $j \in \mathbb{Z}$, $k \in \mathbb{Z}$, and $i \in \{1, 2, 3\}$, we have

$$\psi^i_{j,k}(t) = 2^{-j}\psi(\frac{t - 2^j k}{2^j}). \tag{7.49}$$

As stated in Section 7.2, we will not carry the distinctions between the detail coefficients $i = 1, 2, 3$ in the wavelet decomposition of an image in our presentation, to avoid unnecessary complexity and will refer to $\psi_{j,k}$ in the following. A complete presentation can be found in [4].

For $\Omega \subset \mathbb{R}$, we define the wavelet spaces $W_j := \overline{\text{span}\{\psi_{j,k}\}_{k\in\mathbb{Z}}}$ for $j \in \mathbb{Z}$, where the closure is with respect to $L^2(\Omega)$. The space $L^2(\Omega)$ can be decomposed as the direct sum of the W_j: $L_2(\Omega) = \overline{\bigoplus_{j\in\mathbb{Z}} W_j}$.

Multiresolution analysis is a method for $L^2(\Omega)$-approximation of functions with arbitrary precision. MRA gives approximations on different scales in such a way, that an approximation on a fine scale can be obtained by adding the "details" to an approximation on a coarse scale. The sequence of subspaces defined as $V_i := \oplus_{j\geq i-1} W_j$ forms an MRA such that

$$L_2(\Omega) = \overline{V_i \oplus \bigoplus_{j=i}^{\infty} W_j}. \tag{7.50}$$

Approximation spaces V_i are generated by a scaling function ϕ, constructed using the same dilatation/translation operation of Equation (7.49) [4]. W_j is also the orthogonal complement of V_j inside V_{j+1}:

$$V_{j+1} := V_j \oplus W_j. \tag{7.51}$$

This structure can be used to introduce the decomposition of an image x into different resolution levels. At resolution level $J \in \mathbb{N}$, x has $N = 2^{2^J}$ pixels. A decomposition of $L_2(\Omega)$ at this resolution may thus be obtained as:

$$L_2(\Omega) = V_J \oplus V_J^{\perp} = V_J \oplus \overline{\left(\bigoplus_{j=J}^{\infty} W_j\right)} \tag{7.52}$$

In the following we assume that the image x we want to decompose lives in a resolution J so that $x \in V_J$, which is a standard, and implicit, assumption in the literature (while not being true in general). Such decomposition is valid in a wavelet approximation space if and only if x represent the coefficients of an image projected inside this wavelet approximation space which is a subspace of $L^2(\mathbb{R}^2)$. It holds when the scaling and wavelet functions are both generated by the Haar wavelet [4, 218].

Assume that we want to decompose x until the resolution $J - L$ with $L \in \{1, \ldots, J\}$. As we can decompose V_J into sub-spaces V_{J-1} and W_{J-1}, formally x would be written as:

$$x = \sum_k \langle x, \phi_{J-L,k}\rangle \phi_{J-L,k} + \sum_{j=J-L}^{J-1} \sum_k \langle x, \psi_{j,k}\rangle \psi_{j,k} \tag{7.53}$$

The result of the scalar products are known as [4]

- the approximation coefficients $a_{J-L,k} = \langle x, \phi_{J-L,k} \rangle$,
- and the detail coefficients: for all $j \in \{J-L, \dots, J-1\}, d_{j,k} = \langle x, \psi_{j,k} \rangle$.

The coefficients of the resolution $j-1$, given the approximation coefficients of the resolution j can be obtained through filtering and dyadic sub-sampling:

- the approximation coefficients $a_j = (a_{j+1} * \text{low}) \downarrow 2$,
- and the detail coefficients: $d_j = (a_{j+1} * \text{high}) \downarrow 2$.

$*$ denotes the convolution and $\downarrow$ the dyadic sub sampling (conversely, $\uparrow$ will denote the dyadic up sampling). low and high are a low-pass and a high-pass filter obtained from the wavelet functions (see [219, Chapter 2] or [4] for more details). From the same filters one can construct the inverse operation:

$$a_{j+1} = (a_j \uparrow 2) * \text{low} + (d_j \uparrow 2) * \text{high} \tag{7.54}$$

At a given resolution $j+1$, we will write as Π_{V_j} and Π_{W_j} the projections onto the spaces V_j and W_j respectively. Hence, if $a_{j+1} \in V_{j+1}$, then it admits the following unique representation $\Pi^*_{V_j} a_j + \Pi^*_{W_j} d_j$ where $a_j = \Pi_{V_j} a_{j+1}$ and $d_j = \Pi_{W_j} a_{j+1}$ [4].

Definition of the blocks using MRA. Recall that we want to solve Problem (7.48) for an image x with $N = 2^{2^J}$ pixels. The wavelet decomposition defining the regularization will be done on L-levels. In order to design our block or multilevel algorithm, we need to properly define the blocks, using this wavelet decomposition[6]. At each level, the approximation coefficients will be decomposed in a low frequency block and a high frequency block. Formally, we will define the blocks as follows:

Definition 37. ***Approximation spaces & operators.*** *For all $\ell \in \{J-L+1, \dots, J\}$ we define a pair of filtering-subsampling operators $\Pi_{V,\ell-1}$ and $\Pi_{W,\ell-1}$ such that:*

- $\Pi_{V,\ell-1} : V_\ell \to V_{\ell-1}$,
- $\Pi_{V,\ell-1} : V_\ell \to W_{\ell-1}$,

with V_ℓ and W_ℓ being orthogonal to each other so that $(\Pi_{V,\ell-1})^* \times \Pi_{W,\ell-1} = (\Pi_{W,\ell-1})^* \times \Pi_{V,\ell-1} = 0$ for all $\ell \in \{1, \dots, L\}$. $\Pi_{V,\ell}$ and H_ℓ are projection operators.

We adopt the usual convention that the fine variable x is equal to a_J, meaning that x lives in the approximation space V_J. Every coarse level variable, whether it be high or low frequency components, can be obtained from the fine variable x as follows:

[6] Following our 2-Levels example from Section 7.2.

Lemma 25. *Suppose that for all $\ell \in \{J-L+1,\dots,J\}$ we have a pair of filtering-subsampling operators $\Pi_{V,\ell-1}$ and $\Pi_{W,\ell-1}$ such that:*

$$a_{\ell-1} = \Pi_{V,\ell-1} a_\ell \in V_{\ell-1} \text{ and } d_{\ell-1} = \Pi_{W,\ell-1} a_\ell \in W_{\ell-1} \tag{7.55}$$

Then for all $\ell \in \{J-L+1,\dots,J\}$, one has:

$$a_\ell = \left(\prod_{i=0}^{J-1+\ell} \Pi_{V,\ell+i} \right) x \text{ and } d_\ell = \Pi_{W,\ell} \left(\prod_{i=1}^{J-1+\ell} \Pi_{V,\ell+i} \right) x \tag{7.56}$$

Proof. When writing

$$a_\ell = \left(\prod_{i=0}^{J-1+\ell} \Pi_{V,\ell+i} \right) x, \tag{7.57}$$

we mean

$$a_\ell = \Pi_{V,\ell} \Pi_{V,\ell+1} \dots \Pi_{V,J-1} x. \tag{7.58}$$

The proof is thus straightforward by induction. □

We can also reconstruct each coarse variable with the following relationship with its own coarse variables (i.e., lower levels variables):

Lemma 26. *For all $\ell \in \{J-L+1,\dots,J\}$ we have a pair of projection operators $\Pi_{V,\ell-1}$ and $\Pi_{W,\ell-1}$ such that:*

$$a_\ell = \Pi^*_{V,\ell-1} a_{\ell-1} + \Pi^*_{W,\ell-1} d_{\ell-1} \tag{7.59}$$

Then for all $\ell \in \{J-L+1,\dots,J\}$, one has:

$$\begin{aligned} a_\ell = \Pi^*_{V,\ell-1} \left[\left(\prod_{i=2}^{L-J+\ell} \Pi^*_{V,\ell-i} \right) a_{J-L} + \sum_{i=2}^{L-J+\ell} \left(\prod_{j=2}^{i-1} \Pi^*_{V,\ell-j} \right) (\Pi_{W,\ell-1-i})^* d_{\ell-i} \right] \\ + \Pi^*_{W,\ell-1} d_{\ell-1} \end{aligned} \tag{7.60}$$

Proof. Here the product of operators is to be understood as a composition from left (the first term of the product) to right (last term of the product). The second coarsest variable is written as:

$$a_{J-L+1} = \Pi^*_{V,J-L} a_{J-L} + \Pi^*_{W,J-L} d_{J-L} \tag{7.61}$$

with the coarsest variable being a_{J-L}. The rest follows by induction. □

From these two construction/reconstruction one can define the variable a_ℓ as the coefficients of its own coarse levels: $a_\ell := [a_{J-L}, d_{J-L}, d_{J-L+1}, \dots, d_{\ell-1}]$ for $\ell \geq J-L+2$. This is an abuse of notation, but it is for the sake of conciseness and clarity.

The problem in block form. Now that we are equipped with a definition of the blocks for an L-levels wavelet decomposition, we can rewrite Problem (7.48) so that the blocks directly appear. Recall that we assumed that $a_J := x$. Anticipating on the construction of the L-levels algorithm, we index the fine level problem with its resolution J.

$$\min_{a_J \in \mathcal{H}} F_J(a_J) := \frac{1}{2}\|\mathrm{A}a_J - z\|^2 + \sum_{\ell=J-L}^{J-1} \lambda_\ell \|d_\ell\|_1 \tag{7.62}$$

with $a_L := [a_{J-L}, d_{J-L}, d_{J-L+1}, \dots, d_{J-1}]$. To simplify the presentation, we consider the same problem written as:

$$\min_{a_J \in \mathcal{H}} F_J(a_J) := f_J(\mathrm{A}a_J - z) + \sum_{\ell=J-L}^{J-1} \lambda_\ell g_\ell(d_\ell) \tag{7.63}$$

where we assume the following:

Assumption 8.

- $f_J : \mathcal{H} \to]-\infty, +\infty]$ *is a proper, lower semi-continuous function.*
- $\mathrm{A} : \mathcal{H} \to \mathcal{H}$ *is a linear operator.*
- $\forall \ell \in \{J-L, \dots, J-1\}, g_\ell : W_\ell \to]-\infty, +\infty]$ *is a proper, lower semi-continuous, and proximable function.*

7.5.2 L-levels algorithm for wavelet deblurring

In this section, we construct an L-levels algorithm for Problem (7.62) and check that it is indeed a block-coordinate descent algorithm. The idea is to write explicitly, like in our example of Section 7.2, the iterations of our multilevel algorithm and compare these iterations to that of a block-coordinate descent algorithm.

Coarse level variables. We will consider that at level ℓ the algorithm will update the variable $a_\ell = [a_{J-L}, d_{J-L}, \dots, d_{\ell-1}]$, which amounts to computing parallel updates on the blocks $[a_{J-L}, d_{J-L}, \dots, d_{\ell-1}]$.

Coarse levels functions. Let us first define the coarse level operators A_ℓ for all $\ell \in \{J-L, \dots, J-1\}$ which are Galerkin approximations of the fine level operator $\mathrm{A}_J := \mathrm{A}$.

Lemma 27. ***Coarse level operators.*** *Suppose that Assumption 8 holds. For all $\ell \in \{J-L, \dots, J-1\}$, suppose that we have the following (Galerkin) relationship between the coarse operator A_ℓ and its finer level counterpart $\mathrm{A}_{\ell+1}$:*

$$\mathrm{A}_\ell := \Pi_{V,\ell} \mathrm{A}_{\ell+1} \Pi_{V,\ell}^*, \tag{7.64}$$

and that

$$z_\ell := \Pi_{V,\ell} z_{\ell+1}. \tag{7.65}$$

Therefore, for all $\ell \in \{J-L, \dots, J-1\}$, we have $\mathrm{A}_\ell : V_\ell \to V_\ell$ and

$$\mathrm{A}_\ell = \left(\prod_{i=0}^{J-(\ell+1)} \Pi_{V,\ell+i} \right) \mathrm{A} \left(\prod_{i=1}^{J-\ell} \Pi_{V,J-i}^* \right), \tag{7.66}$$

and

$$z_\ell = \left(\prod_{i=0}^{J-(\ell+1)} \Pi_{V,\ell+i} \right) z. \tag{7.67}$$

Proof. Proof is in Appendix A.6. □

We can now define the coarse level functions as follows.

Definition 38. ***Coarse level functions.*** *For all $\ell \in \{J-L+1, \dots, J-1\}$, the function associated with the coarse level ℓ is defined as:*

$$F_\ell(a_\ell) := f_\ell(\mathrm{A}_\ell a_\ell - z_\ell) + \sum_{j=J-L}^{\ell-1} \lambda_j g_j(d_j) + \langle v_\ell, \cdot \rangle, \tag{7.68}$$

where v_ℓ enforces the first order coherence between levels ℓ and $\ell+1$. Each A_ℓ can be expressed from A using Lemma 27.

Remark 23. *The gradient of the data fidelity term can be expressed using Lemma 27 as:*

$$\nabla f_\ell = \left(\prod_{i=0}^{J-(\ell+1)} \Pi_{V,\ell+i} \right) \mathrm{A}^* \left(\mathrm{A} \left(\prod_{i=1}^{J-\ell} \Pi_{V,J-i}^* \right) \cdot - \left(\prod_{i=1}^{J-\ell} \Pi_{V,J-i}^* \right) z_\ell \right) \tag{7.69}$$

This expression will help us compute the correction term v_ℓ in the next section.

Coarse level corrections: enforcing first order coherence. We now compute explicitly the correction term v_ℓ for all level ℓ. After that, we will be able to write the iterations of the L-levels algorithm and compare them to that of a block-coordinate descent algorithm. First, we look at the first order coherence between levels ℓ and $\ell+1$ (without considering upper levels dependencies).

Lemma 28. *Suppose that Assumption 8 holds and that coarse models are defined as in Definition 38. For all $\ell \in \{J-L+1, \dots, J-1\}$, the correction term v_ℓ enforcing the first order coherence between levels ℓ and $\ell+1$ is given by:*

$$v_\ell = \Pi_{V,\ell} \mathrm{A}_{\ell+1}^* \left(\mathrm{A}_{\ell+1} \Pi_{W,\ell}^* \Pi_{W,\ell} \left(a_{\ell+1} - z_{\ell+1} \right) \right) + \Pi_{V,\ell} v_{\ell+1}. \tag{7.70}$$

Proof. Proof is in Appendix A.6. □

Remark 24. *One can recognize in the first part of v_ℓ the first order coherence term we computed in the motivating example of Section 7.2.*

A brief development then yields the following lemma.

Lemma 29. *Suppose that Assumption 8 holds and that coarse models are defined as in Definition 38. For all $\ell \in \{J-L+1, \dots, J-1\}$, the correction term v_ℓ enforcing the first order coherence between levels ℓ and $\ell+1$, with respect to level J, is given by:*

$$v_\ell = \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \mathrm{A} \left(\sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi_{V,J-k}^* \right) \Pi_{W,\ell+i-1}^* d_{\ell+i-1} \right) \tag{7.71}$$

$$- \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi_{V,J-k}^* \Pi_{W,\ell+i-1}^* \Pi_{W,\ell+i-1} \left(\prod_{k=0}^{J-\ell-i} \Pi_{V,\ell+i+k} \right) z \right). \tag{7.72}$$

Proof. Proof is in Appendix A.6. □

We can therefore express every iteration of our multilevel algorithm using only A and z as follows. One pass of gradient descent on the smooth component of level ℓ reads:

$$\begin{aligned} \nabla f_\ell(a_\ell) + v_\ell = & \left(\prod_{k=0}^{J-(\ell+1)} \Pi_{V,\ell+k} \right) \mathrm{A}^* \mathrm{A} \left(\prod_{k=1}^{J-\ell} \Pi_{V,J-k}^* \right) a_\ell \\ & + \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \mathrm{A} \left(\sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi_{V,J-k}^* \right) \Pi_{W,\ell+i-1}^* d_{\ell+i-1} \right) \\ & - \left(\prod_{k=0}^{J-(\ell+1)} \Pi_{V,\ell+k} \right) \mathrm{A}^* \left(\prod_{k=1}^{J-\ell} \Pi_{V,J-k}^* \right) z_\ell \\ & - \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi_{V,J-k}^* \Pi_{W,\ell+i-1}^* \Pi_{W,\ell+i-1} \left(\prod_{k=0}^{J-\ell-i} \Pi_{V,\ell+i+k} \right) z \right) \end{aligned} \tag{7.73}$$

One can identify in the first two lines the contribution of all the wavelet coefficients $[a_\ell, d_\ell, \dots, d_{J-1}]$ to the gradient of the smooth component of level ℓ; while the last two lines contain the contribution of the wavelet coefficients of z from decomposition level ℓ to $J-1$.

Block updates. We now express the block updates in the same way as the multilevel algorithm. As we have seen in Section 7.2, computing the contribution of the gradient of the data fidelity term to the block in question is the main difficulty. We will assume, as our H-BC-FB algorithm can handle parallel updates, that we can update blocks a_ℓ for $\ell \in \{J-L+1, \dots, J-1\}$, which are in practice groups of blocks as:

$$x := a_J = [a_{J-L}, d_{J-L}, d_{J-L+1}, \dots, d_{J-1}].$$

In the following, we will refer to these groups of blocks as blocks to simplify the presentation. For all $\ell \in \{J-L, \dots, J-1\}$, the gradient of the data fidelity term to the block a_ℓ is denoted by $\nabla_\ell f_J$ and is given by:

$$\nabla_\ell f_J(a_J) = \left(\prod_{i=0}^{J-(\ell+1)} \Pi_{V,\ell+i} \right) \mathrm{A}^* \left(\mathrm{A} a_J - z \right) \tag{7.74}$$

Therefore we have the following lemma.

Lemma 30. *Suppose that Assumption 8 holds. One iteration of a multilevel algorithm with L levels at level ℓ, for all $\ell \in \{J-L, \dots, J\}$, is equivalent to one iteration of a block-coordinate algorithm updating blocks $[a_J, d_J, \dots, d_{\ell-1}]$.*

Proof. The result comes directly from the expression of the gradient of the data fidelity term with respect to $a_\ell = [a_J, d_J, \dots, d_{\ell-1}]$ (Equation (7.74)) and the expression of the gradient of the smooth component of level ℓ (Equation (7.73)).

The proximal step is straightforward. □

With this last lemma we have established the equivalence between the multilevel algorithm and a block-coordinate algorithm. We can now analyze our multilevel algorithm with this lens.

7.5.3 What we learned from the BC point of view

In this section, we present what we learned by looking at multilevel algorithms through BC glasses.

Coarse levels, coarse spaces, and information transfer operators. At the beginning of our work on multilevel algorithms, and several times since then, we asked ourselves and were asked if using wavelet based information transfer operators to solve our wavelet deblurring problem could help the analysis of the algorithm. It turns out it can. Indeed, when looking at our problem through the BC lens, it appears natural to define coarse levels so that they respect the hierarchy the wavelet regularization induces on the solution.

Hence, multilevel algorithm should adopt the hierarchy induced by the regularization, if it exists, or at least the separability of the problem.

Along the various numerical experiments we conducted, we also remarked that constructing the coarse levels from the wavelet bases of the regularization was beneficial to the convergence of the algorithm - even if what we learned to really matter was more the length of the wavelet filters than the wavelet basis per se (e.g. wavelets known as Daubechies 20 and Symlet 10 display similar results in practice).

Non-smooth fine level implies non-smooth coarse levels. Just as with the selection of coarse spaces, the non-smoothness of the fine level function should prompt the non-smoothness of the coarse levels functions. This was something we did not observe in practice with IML FISTA between the smooth and non-smooth coarse models on the acceleration of the convergence (see Section 5.4, Chapter 5): the non-smooth coarse model had a small advantage over the smooth one on high noise settings, but otherwise the performance was similar.

However, these experiments were conducted without knowing that to compute the first order coherence term, when choosing non-smooth coarse models, we did not need to compute the gradient of the smoothed regularization (as they cancel each other between fine and coarse levels). This is a significant advantage of non-smooth coarse models that we did not exploit.

First order coherence is necessary. Another important question was the necessity of the first order coherence term. Computing the first order coherence is without a doubt the most computationally expensive part of a multilevel iteration. Thus, we were wondering if it could be avoided. The answer seems to be no: the first order coherence term is necessary to bridge multilevel algorithms and BCD algorithms.

From a practical point of view, we observed in some tests that the first order coherence term was paramount in the good behavior of IML FISTA. Now, from a theoretical point of view, we showed that the first order coherence term is bridging the gap between a convergent BC FB algorithm and our multilevel algorithm, as illustrated in Equation (7.73).

Minimizing coarse levels completely is useless in general. This BC interpretation comes with another important conclusion: minimizing the coarse levels completely is useless in general. Indeed, as first order coherence is sending high frequency information to the coarse levels at the beginning of the coarse optimization, minimizing completely the coarse levels functions would mean that higher frequency information would not influence coarser level in any way after this initial correction. This is untrue in general, and we use our example of Section 7.2 to show it.

The coarse level operator A_H is obtained as a Galerkin approximation of A on the low frequency block: $\mathrm{A}_H = \Pi_V \mathrm{A} \Pi_V^*$. An iteration on the coarse level is then given by:

$$a_{n+1} = \mathrm{prox}_{\gamma \lambda_H g_H} \left(a_n - \gamma \Pi_V \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* a_n + \Pi_W^* d_n \right) - z \right) \right) \tag{7.75}$$

For minimizing completely our objective function with respect to $a \in V$ to be useful, we would need the following condition to hold for all $d \in W$:

$$\Pi_V \mathrm{A}^* \mathrm{A} \Pi_W^* d = 0, \tag{7.76}$$

which is quite strong (and was not true in our experiments). Hence, in general, minimizing completely the coarse levels is not useful. It is also showing that fine level steps are paramount for the convergence of a multilevel algorithm.

7.6 Numerical experiments

In this section, we present preliminary numerical experiments to assess the performance of the proposed construction of a multilevel algorithm. We have seen in Chapter 5, and notably in Section 5.4 that multilevel algorithms do accelerate the solution of Problem (7.48) (see Figure 5.2 in particular).

Dataset. In this section we will consider two versions of the first image of the JWST (see Figure 2.2, Chapter 2), one of size 512×512 and one of size 2048×2048. We will apply a Gaussian blur and a Gaussian noise on these images. The regularization will be done with a 2-Level ℓ_1-Wavelet Symlet 10 (see Figure 7.1).

Experimental setup. We will compare our algorithm to several versions of BC-FB and to the forward-backward algorithm. We will consider the following algorithms:

- FB: the forward-backward algorithm.
- Random BC-FB: the block-coordinate forward-backward algorithm with one randomly chosen block updated at each iteration. The blocks correspond to the approximation block and the three detail blocks.
- Random parallel BC-FB: the block-coordinate forward-backward algorithm with one or more randomly chosen blocks updated at each iteration, in parallel. Same block structure as above.
- IML FB (H-BC-FB): the multilevel algorithm with 2 levels, with m coarse iterations every 10 fine iterations.
- Stochastic H-BC-FB: a random multilevel algorithm with 2 levels, with m coarse iterations every 10 fine iterations in expectation. It is based on a stochastic and parallel BC-FB algorithm, with the selection rule of Lemma 24.

For all block algorithms, the number of blocks will be 4, following the wavelet decomposition (see Figure 7.1); at the same time, multilevel algorithm will always update all the details coefficients at the same time. With this choice we intend to show that our block update rule, which forces the update of the approximation coefficients at each iteration, is more efficient than a random one. Note that we also tested a Random BC-FB algorithm that splits the image in four equally sized patches, but the results were worse than all the other algorithms presented here.

Implementation constraints. The efficiency of BCD algorithms on problems of the form (7.62) is highly dependent on the fact that the operator A can be applied at the block scale. It is possible in this context to pre-compute the operator A on the blocks, i.e., to compute in advance $\Pi_V^* \mathrm{A}^* \mathrm{A} \Pi_V$ and $\Pi_W^* \mathrm{A}^* \mathrm{A} \Pi_W$. However, we did not have the time to implement this pre-computation efficiently, therefore BCD and multilevel algorithms are at a disadvantage compared to FB.

Comparison of several update choices. A first test to evaluate the interest of our method is to vary the number of coarse iterations m in the multilevel algorithm, therefore looking at the weight we put on updating the coarse level with respect to the detail coefficients.

The setting is the following for the multilevel algorithms: every 10 iteration we compute m coarse iterations, with m varying from 1 to 10: for a standard multilevel algorithm, setting $m = 1$ consists in applying one iteration of the projected fine level gradient, like applying a low frequency approximation of the gradient at fine level; setting $m = 10$ shows what happens if we only update the approximation coefficients. For our random multilevel algorithms we implement in expectation the same behavior, with a small caveat: we stopped at $m = 9$ in order not to put the probability of a fine level iteration to 0 (if $m = 10$);

We can see in Figure 7.4 that the multilevel algorithm with $m = 1$ is already outperforming FB and all BC-FB algorithms. There seems to be a sweet spot around $m = 5$ where the convergence is the fastest, but the difference with other versions of the algorithm is marginal, therefore we only displayed this configuration. The main difference is at the start of the optimization, where multilevel algorithms with m big enough seems slower than FB but quickly overtakes it, which indicates that the coarse level contains information that is then useful to accelerate the optimization.

We expected that the BC-FB algorithms would be slower than FB, given the implementation constraints we mentioned earlier. It is interesting to note that the random BC-FB algorithm is performing better than the random parallel BC-FB algorithm, even though the latter exploits more information at each iteration. Both algorithms are vastly outperformed by the multilevel algorithm and its random version, which not only corroborate our previous findings but also show that clever selection of the blocks to update can lead to faster convergence.

To be completely fair to BC algorithms, a multilevel rule to update the blocks still require updating all the blocks at once, every few iterations. Therefore, standard update rule are still relevant when the problem is big enough that updating all the blocks at once is no longer possible.

High dimensional test. We conclude this experiment section with a test of our approach on a high dimensional problem. We consider a 2048×2048 image of the JWST (see Figure 2.2, Chapter 2) and apply the same type of degradation as in the previous experiment. The results are displayed in Figure 7.5. Again, our algorithm show promising results, beating both forward-backward algorithm and block approaches.

7.7 Conclusion

In this chapter, we proposed a new convergent BC-FB algorithm, based on a hierarchical strategy for the block selection, able to tackle non-convex and non-smooth optimization problem. This proposition was motivated by the new perspective it brings to analyze multilevel algorithms. In the case of image deblurring with ℓ_1-wavelet regularization, we followed the standard approach to construct a multilevel algorithm to tackle this problem. We have shown that this algorithm can be viewed as a block-coordinate descent algorithm, which allowed us to analyze its convergence properties and more notably to

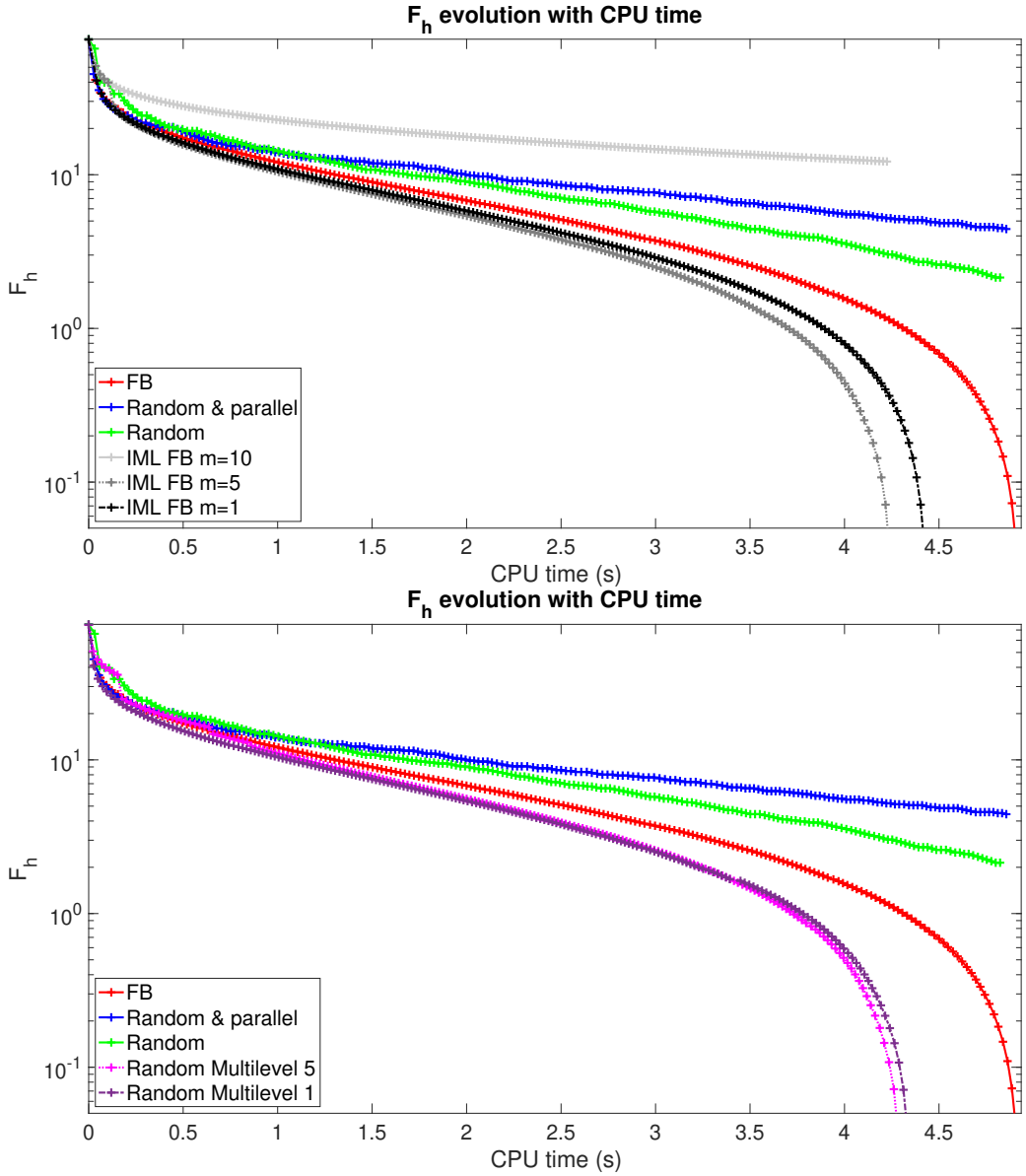

Figure 7.4: Comparison of the convergence of multilevel algorithms against FB (red), random BC-FB (green, one block updated at each iteration), and random parallel BC-FB (blue) for the deconvolution problem regularized with 2-Level ℓ_1-Wavelet Symlet 10 on a 512×512 image of the JWST (see Figure 2.2, Chapter 2). Degradation: Gaussian noise with $\sigma_{\text{noise}} = 0.05$ and a Gaussian blur of size 10×10 and 1.8 standard deviation. $\lambda_a = 1 \times 10^{-7}$, $\lambda_d = 1 \times 10^{-2}$. (Top) IML FB: standard multilevel algorithm: every 10 iteration we compute m coarse iterations (m being indicated in the legend). (Bottom) Random Multilevel: same behavior but in expectation (except that for $m = 10$ we did not put the probability of a fine level iteration to 0).

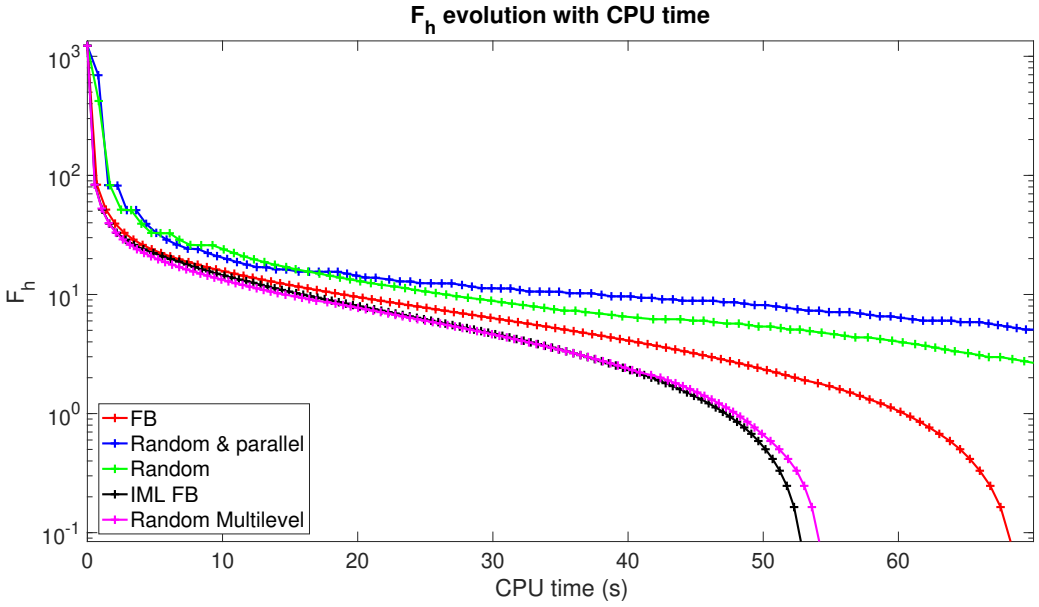

Figure 7.5: Comparison of the convergence of multilevel algorithms against FB (red), random BC-FB (green, one block updated at each iteration), and random parallel BC-FB (blue) for the deconvolution problem regularized with 2-Level ℓ_1-Wavelet Symlet 10 on a 2048×2048 image of the JWST (see Figure 2.2, Chapter 2). Multilevel: standard multilevel algorithm: every 10 iteration we compute 2 coarse iterations. Random Multilevel: same behavior but in expectation. Degradation: Gaussian noise with $\sigma_{\text{noise}} = 0.05$ and a Gaussian blur of size 40×40 and 7.7 standard deviation. $\lambda_a = 1 \times 10^{-7}$, $\lambda_d = 5 \times 10^{-2}$.

understand the importance of each build block of a multilevel algorithm, e.g. the first order coherence. Finally, we conducted preliminary numerical experiments that demonstrate the effectiveness of our approach compared to traditional BC-FB algorithms.

We believe that this new perspective will help us to further improve the performance of multilevel algorithms. A promising first direction is the design of multilevel algorithm with *a priori* rules, i.e., rules that would not require an extensive tuning of hyperparameters. We could for instance revisit the image restoration problem we treated in Chapter 5 and in Chapter 6 to see if we can improve the performance of our algorithms by following the BC perspective we developed in this chapter.

Conclusion

Conclusion

In this thesis we present an in-depth study of multilevel methods for non-smooth optimization, with application to image reconstruction problems. By exploiting a hierarchy of approximations of the objective function, multilevel algorithms can accelerate the convergence of optimization algorithms.

In Chapter 4, we presented a general framework for designing a multilevel algorithm with state-of-the-art convergence guarantees for optimization problems that may be non-smooth, and without an explicit formulation of their proximity operators. Specifically, a convergence rate of $O(1/k^2)$ for the objective function values and convergence to a minimizer. This framework led to IML FISTA, an inexact multilevel variant of FISTA.

With this algorithm we have been able to show that we can accelerate the solution of image reconstruction problems, with a speedup of up to 10 times compared to the state-of-the-art algorithms. In Chapter 5, we applied our algorithms to the following problems:

(i) grayscale image deblurring regularized with wavelet transform,

(ii) color image deblurring regularized with total variation,

(iii) color image inpainting regularized with non-local total variation,

(iv) hyperspectral image inpainting and deblurring regularized with non-local total variation,

and showed that it can provide a good acceleration on all of them. From these experiments, we thus concluded that multilevel methods are valuable approaches to accelerate the solution of image reconstruction problems, and that they can be applied to a wide range of problems.

This is why we decided to tackle a more realistic imaging problem in Chapter 6: radio-interferometric (RI) image reconstruction regularized with wavelet transform and positive constraints. We showed that IML FISTA can be successfully applied to this problem, providing a significant speedup compared to the state-of-the-art algorithms. Moreover, we implemented a new way of reducing the dimension of the problem, by constructing our hierarchy of levels based on a selecting fewer and fewer observations (visibilities) at each level, to be more in tune with the scaling challenges of RI problems [169].

We also remarked that the theoretical guarantees of IML FISTA are still not on par with the performance we can observe in practice, which is a good sign for the robustness of the method, but also an incentive for more theoretical work. Up to this point, our experience with IML FISTA is consistent with the experience of other authors on multilevel methods[7], that is that the primary difficulty in getting them work lies in the implementation details: choice of coarse models, choice of algorithms at each level, and so on.

To illustrate this phenomenon with a specific example, in the general context of IML FISTA, we could only guarantee that a multilevel step would decrease the fine level objective function up to a small error. In our experiments we never observed this error, which prompts the question: is it possible to prove that the multilevel step is always a descent step?

Progress in this direction (and others) was obtained in Chapter 7, where we studied the convergence of multilevel algorithms from the point of view of block-coordinate descent algorithms. For this study, we provided a convergence proof for a new hierarchical block-coordinate forward-backward algorithm, applicable to non-smooth and non-convex optimization. Regarding our question, in this context, we are able to guarantee that a multilevel step is a descent step, even for non-convex problems.

This new point of view led us to a rigorous design of a multilevel algorithm for image deblurring regularized with wavelets, that solved all the theoretical issues we encountered with IML FISTA in this context.

The work presented in this manuscript opens the door for new theoretical and practical developments.

Practical perspectives

First from an application perspective, our numerical experiments further highlighted that multilevel methods have a great potential on imaging problems where a sequence of optimization problems needs to be solved. On such applications, the gain of the multilevel approaches could be compounding, leading to even greater speedups. For instance, in RI imaging, the volume of data to handle is so large that it is common to reconstruct the image by solving a sequence of optimization problems with disjoint set of observations, in an online fashion [186]. The multilevel approach could be used to accelerate the resolution of each individual problem. To continue with RI imaging, benchmarking the performance of IML FISTA on real data would be a natural next step. Then we could consider extending our construction of multilevel algorithms to tackle the constrained version of SARA (see Appendices A.3.2 and A.2.3).

From an image restoration perspective, there are also plenty of directions that could be explored. To start with the variational approaches, we talked briefly about Total Generalized Variation (TGV) in Chapter 2. It offers a more flexible regularization than TV, and is simpler than NLTV while providing similar reconstruction results. TGV is not immediately handled by our framework of multilevel algorithms, as it is not formulated as the composition of a norm and a linear operator [34], and could be an interesting extension.

[7] As noted in Chapter 3, Section 3.2.2.

With the introduction of deep learning techniques for image restoration, it would be interesting to see how they could be combined with multilevel algorithms. The recent and numerous developments to train deep neural networks to mimic variational regularizers such as RED [39, 40] or Plug-and-Play (PnP) methods [38] seems to be a good starting point. RED for instance has a smooth formulation [39], which consists in writing the regularization as

$$R(x) = g_\sigma(x), \tag{7.77}$$

with its gradient being explicitly formulated as: $\nabla g_\sigma : x \mapsto x - \mathrm{NN}_\sigma(x)$, where NN_σ is a denoising neural network parametrized by a noise level σ. PnP methods, on the other hand, links the neural network to the proximity operator of the regularization, so that $\mathrm{prox}_{g_\sigma} = \mathrm{NN}_\sigma$ [41–43]. In both cases, the resulting method is an iterative one, with a great cost with respect to variational approaches as evaluating a neural network is computationally expensive [220][8]. Hence, a multilevel algorithm could help alleviate this cost. However, we have seen in our own experiments that the coarse models need to be efficient to provide a speedup, therefore we need to construct a clever approximation of the neural network, or for instance, not regularize the coarse level like in Chapter 6 (while maintaining first order coherence). Some work is currently being done in this direction.

The last chapter also suggests revisiting some of our numerical experiments, in particular the application to hyperspectral image restoration. The notion of band is analogous to the notion of blocks in our BC theoretical framework. It could be interesting to define a multilevel algorithm that would exploit the intrinsic hierarchy of the bands [223, 224], and thus maybe improve the performance of the algorithm.

Following similar principles on new optimization problems to construct multilevel algorithms beforehand, could help skip a lot of the trial-and-error process that is currently needed to design a multilevel algorithm on a new problem.

Theoretical perspectives

For IML FISTA. From a theoretical perspective, even if we managed to make some progress in the convergence analysis of multilevel algorithms in Chapter 7, a gap remains with the generality of our framework IML FISTA.

First, coarse corrections are guaranteed to decrease the objective function in the BC framework, that is not guaranteed in the IML FISTA framework (or under any other general multilevel framework for non-smooth optimization). We tried to investigate improvement of the smoothing framework in Appendix A.2.1, by looking at a sufficient decrease condition, but the results are not exploitable in practice. Hence, the question remains open.

Second, inexact proximal steps as characterized by [62] in the convex case, are not possible (yet) in the Kurdyka-Łojasiewicz setting, even though some notion of inexactness already exist [212, 214]. Trying to extend our convergence results obtained in Chapter 7 to this setting would be a natural next step.

[8] To quote the authors: "The inference of neural networks is claimed to represent 90% of the cost of machine learning at scale according to independent reports from both NVIDIA [221] and Amazon Web Services [222]."

Finally, on a more positive note, the proof of convergence we used in Chapter 4 can be reused to prove the convergence of other multilevel algorithms, as we demonstrate it in Appendix A.3.2. Therefore, we can concentrate on finding efficient implementations of these algorithms, without worrying about having to prove convergence.

For BCD. Now from our BC point of view, one can wonder if the results of Chapter 7 could be proven in a complete (non-convex) stochastic setting. Is it possible to consider correlations between updates from one iteration to the next and still obtain the same guarantees as in the Féjer setting [196]? Answering this question would greatly improve convergence guarantees of some stochastic BC methods.

It was proven in [179] that the reweighting procedure involved in SARA [167] could be interpreted as solving a unique optimization problem with a block-coordinate descent algorithm, and later applied to RI imaging [181]. Our new block-coordinate perspective could be used to analyze the convergence of a multilevel algorithm applied to this problem.

Also, it is known that BC algorithms are really competitive in settings where updating all the coordinates at once is not possible [192]. Hence, it would be interesting to see how we could adapt our hierarchical selection of the blocks to update in such contexts.

Finally, we did not study the impact of adding inertia [190] on the convergence of our H-BC-FB algorithm. Such study would also help us end the analogy between IML FISTA and H-BC-FB, and maybe provide some insights on how to improve the convergence of IML FISTA: should we use inertia in the multilevel steps?

Higher order optimization. Higher order optimization methods are known to adapt better to the geometry of the function and therefore converge faster, but at a higher cost [55, 225]. First order methods are more understood, and more used than higher order methods, and their potential is, most likely, close to be fully exploited nowadays. Therefore, some effort should be made to reduce the computational cost of higher order optimization methods.

From a non-smooth multilevel optimization perspective, a first step in this direction could be taken by trying to emulate the second order coherence with the data fidelity, constructing a Galerkin approximation of the Hessian matrix at coarse level. Such can be implicitly done when choosing $\mathrm{A}_H = I_h^H \mathrm{A}_h I_H^h$ in our image restoration problems, as $\mathrm{A}_H^T \mathrm{A}_H = I_h^H \mathrm{A}_h^T I_H^h I_h^H \mathrm{A}_h I_H^h = I_h^H \mathrm{A}_h^T \mathrm{A}_h I_H^h$ if $I_H^h I_h^H = \mathrm{Id}_H$ (the identity at coarse level). This is trivially satisfied within our BC point of view of multilevel algorithm for wavelet deblurring. Studying the eigenvalues of $I_h^H \mathrm{A}_h^T \mathrm{A}_h I_H^h$ and their relationship to those of $\mathrm{A}_h^T \mathrm{A}_h$ could tell us how faithful to the fine level is the second order information of the coarse level, without having to send the actual Hessian matrix at coarse level at each multilevel coarse correction.

Such ideas could be incorporated into a multilevel proximal Newton algorithm [226] which uses the Hessian matrix of the smooth term in the iterations to better adapt the forward step to the geometry of the function.

Appendix A

A.1 Chapter 3 – Supplementary literature on multilevel algorithms

In this section, we present some literature on multilevel optimization that is not directly relevant to this manuscript but is of interest. First we consider applications of the multilevel framework to either accelerate or improve the training of neural networks.

Deep learning. It seems natural to want to use the multigrid framework to accelerate the training of neural networks as it can be quite expansive. In [227], the authors present a way to define a hierarchy of neural networks inducing thus a hierarchy of losses so that one can accelerate the training of deep neural networks. The idea relies on the fact that deep neural networks known as ResNets can be seen as the discretization of partial differential equation. A layer of ResNet obeys the following equation:

$$y_{k+1} = y_k + \Delta_k F(y_k, \theta_k), \tag{A.1}$$

where $\delta_k = 1$, and $\theta_k = (W_k, b_k)$ represent the parameters of the layer k so that:

$$F(y_k, \theta_k) = \sigma(W_k y_k + b_k),$$

with σ a nonlinear activation function (e.g. rectified linear unit (ReLU)). Equation A.1 can be seen as the discretization of the following ordinary differential equation when Δ_k goes to zero:

$$\begin{aligned} \frac{dy}{dt} &= F(y, t) \\ y(0) &= y_0. \end{aligned} \tag{A.2}$$

Thus, by adjusting the step size Δ_k in the ResNet, we implicitly adjust the mesh size of the discretization. The hierarchy of neural networks is thus naturally defined and can be used to construct a multilevel algorithm.

Authors of [228] rather use the multilevel procedure to regularize the solution of the loss function associated with a deep neural network. This way the method is claimed to avoid overfitting of the neural network to the training dataset.

Miscellaneous multilevel algorithms. The term multilevel may be confusing given that it shares the same name as bilevel optimization field. It can be justified by the fact that the hierarchy of levels may not necessarily be constructed to reduce the dimension of the problem, but rather to reduce its complexity.

For instance the authors of [130] proposed a method to define an approximation of the loss in a deep learning context. The loss is classically defined with respect to a dataset, and the authors proposed to construct another loss to minimize with synthetic data. Let L be the loss we aim to minimize and $\bar{L}$ its approximation. The approximation relies on the following assumptions:

- $h := L - \bar{L}$ is differentiable, and its gradient is δ-Lipschitz continuous.
- This is equivalent to require that for all w, $\|\nabla^2 L(w) - \nabla^2 \bar{L}(w)\|_{\text{op}} \leq \delta$ if L and $\bar{L}$ are twice differentiable.

By construction, it is already the approximation is already first order coherent with L [130] (using the stochastic gradient), and with these assumptions, in a sense, the approximation should be coherent up to second order with the original loss.

Multilevel algorithms have also been used to accelerate the solving of graphical lasso problems [229]. Such problems are graph reconstruction problems regularized with an ℓ_1-norm. Authors of [229] design a hierarchy of sub-problems by restricting the number of variables to be updated at any given iteration. The fact that the ℓ_1-norm provide separable regularization and also identifies the support of the solution (often much smaller than the actual dimension of the problem) justifies the use of this approach. Nonetheless, convergence of the method to a global minimizer requires each coarse model to be solved exactly.

For particular image restoration problem such as Total Variation denoising, explicit construction of all the levels was tried for instance in [230]. The authors present a multilevel algorithm to solve a deblurring problem regularized with a smoothed total variation but in a variational sense. The approach consists in defining the functional for the smallest group of pixels possible (4×4 for TV), minimize it and then upscale the solution to the next level (groups of pixels of size 8×8 and so on). Each group at given scale is solved in parallel. As long as the functional at end is smooth, the method converge with optimal complexity to the solution of the original problem. The main difficulty is to define the correct functional at each scale, which is already quite involving for the simple case of the TV regularization. This kind of idea was previously developed in [231] but for different optimization problems.

Finally, in [232], the authors design a FISTA algorithm able to work by computing iterations in subspaces of different dimensions. This idea is quite interesting and can work in infinite dimensional setting, but the formulation of the algorithm is completely different from the one of multilevel algorithms. There may be some connections to be drawn between IML FISTA and this algorithm, as the extrapolation steps in [232] do not always occur in the "fine" space.

A.2 Chapter 4 – Possible improvements of IML FISTA's framework

In this section, we look at possible improvements of the framework of IML FISTA. In particular, we look at the smoothing technique, and the descent guarantees it produces; the extrapolation step; and conclude with some unaddressed limits of the method.

A.2.1 Improving the smoothing: sufficient decrease and other techniques.

In this section, we discuss the limitations of the framework we use to define the coarse models of IML FISTA in Chapter 4, and the subsequent descent guarantees at fine level, to try to propose some improvements.

Recall that with our method, a coarse correction amounts to minimizing through the coarse model a smooth approximation of the objective function. This results in a descent guarantee up to an error (Chapter 4, Lemmas 10 and 13).

$$F_h(y_h + \bar{\tau} I_H^h(s_{H,m} - s_{H,0})) \leq F_h(y_h) + (\eta_1 + \eta_2)\gamma_h.$$

This error can be made as close to 0 as possible by driving γ_h to 0. However, doing so is detrimental to the practical performance of the algorithm in the applications we considered (see Chapter 5). Thus, a question that we tried to tackle during my PhD was: *Can we guarantee descent of the objective function at the fine level when using a smooth approximation of the objective function?*

In practice, we never observed a non-decreasing objective function, but a theoretical argument would certainly make our method more robust. We present in the following our findings in the most general setting, in order to extend the guarantees we obtained in Chapter 7.

Sufficient decrease condition. The first idea that comes to mind is to decrease our smoothed function sufficiently so that it is guaranteed to then decrease the non-smooth objective function. From the definition of the smoothed coarse models (Chapter 4, Definition 25), we have that for all $x \in \mathcal{H}$:

$$\begin{aligned} F_{h,\gamma_h}(x) &\leq F_h(x) + \eta_2\gamma_h \\ F_h(x) &\leq F_{h,\gamma_h}(x) + \eta_1\gamma_h. \end{aligned}$$

Thus, finding $y \in \mathcal{H}$ such that $F_{h,\gamma_h}(y) \leq F_{h,\gamma_h}(x) - \eta\gamma_h$ guarantees that $F_h(y) \leq F_h(x)$. This leads to the following lemma:

Lemma 31. *Let F be a convex function on a convex set X. Let F_γ be a $\gamma-$smooth approximation of F with $\gamma > 0$. Suppose that for $x \in X$ there exists $y \in X$ such that*

$$F_\gamma(y) \leq F_\gamma(x) - \eta\gamma, \tag{A.3}$$

then we have

$$F(y) \leq F(x). \tag{A.4}$$

Proof. The proof comes from the steps above. □

It remains now to fulfill the strong assumption of this lemma, finding y such that Equation (A.3) is valid. This boils down to finding an algorithm, and a sufficient number of iterations so that this condition is met. In [132], the authors proposed a parametrization of the smoothing, so that with a fast method (with a rate of convergence $1/k^2$) one can estimate the sufficient number of iterations to reach an $\epsilon-$approximation of the minimum value of the objective function. We will place ourselves in a similar setting.

A fast iterative method for convex optimization is given by the following definition:

Definition 39. ***Fast iterative method [132, Definition 3.1].*** *Let (L, R, β) be a given input convex optimization model with an optimal solution $\widehat{x}$, and let $x_0 \in \mathcal{H}$ be an initial point. An iterative method $\mathcal{M}$ for solving problem (2.7) is called a fast iterative method with constant $0 < \Gamma < +\infty$, which possibly depends on x_0 and $\widehat{x}$, if it generates a sequence $(x_k)_{k\in\mathbb{N}}$ satisfying for all $k \geq 1$,*

$$F(x_k) - F(\widehat{x}) \leq \frac{\beta\Gamma}{k^2}. \tag{A.5}$$

With such a method, and the smoothing of R proposed in [132] to define the smooth problem F_γ, the following theorem holds:

Theorem 11. ***[132, Theorem 3.1].*** *Let $(x_k)_{k\in\mathbb{N}}$ be the sequence generated by a fast iterative method $\mathcal{M}$ when applied to the smooth problem F_γ. Let $\epsilon > 0$. Suppose that the smoothing parameter is chosen as:*

$$\gamma = \sqrt{\frac{\alpha}{\eta}} \frac{\epsilon}{\sqrt{\alpha\eta} + \sqrt{\alpha\eta + (\beta + K)\epsilon}}. \tag{A.6}$$

Then for

$$k \geq 2\sqrt{\alpha\eta\Gamma}\frac{1}{\epsilon} + \sqrt{(\beta + K)\Gamma}\frac{1}{\sqrt{\epsilon}}, \tag{A.7}$$

it holds that $F(x_k) - F(\widehat{x}) \leq \epsilon$.

In our context, we will start by working only on F_γ with $\mathcal{M}$:

Lemma 32. *Let $(x_k)_{k\in\mathbb{N}}$ be the sequence generated by a fast iterative method $\mathcal{M}$ when applied to the smooth problem F_γ. Suppose that $F_\gamma(x_0) - F_\gamma(\widehat{x}) > \eta\gamma$. If k is such that:*

$$k \geq \sqrt{\frac{\beta\Gamma}{F_\gamma(x_0) - F_\gamma(\widehat{x}) - \eta\gamma}} \tag{A.8}$$

then

$$F_\gamma(x_k) \leq F_\gamma(x_0) - \eta\gamma. \tag{A.9}$$

Proof. From the definition of $\mathcal{M}$ we have that:

$$F_\gamma(x_k) - F_\gamma(\widehat{x}) \leq \frac{\beta\Gamma}{k^2}. \tag{A.10}$$

Injecting k in this inequality yields

$$F_\gamma(x_k) - F_\gamma(\widehat{x}) \leq \frac{\beta\Gamma}{\frac{\beta\Gamma}{F_\gamma(x_0)-F_\gamma(\widehat{x})-\eta\gamma}} = F_\gamma(x_0) - F_\gamma(\widehat{x}) - \eta\gamma, \tag{A.11}$$

which concludes the proof. □

This lemma is not satisfying, for the simple reason that if one is close enough to a minimizer, $F_\gamma(x_0) - F_\gamma(\widehat{x}) > \eta\gamma$ has no reason to hold ($\eta\gamma$ is of the order of N for the ℓ_1-norm smoothed by the Moreau envelope).

A better result would depend on a relative instead of absolute difference between $F_\gamma(x_0)$ and $F_\gamma(\widehat{x})$. Consider Lemma 7, we have that for all $x \in X$:

$$R(x) - \gamma\omega^*(\mathbf{d}_x) \leq R^{ic}_\gamma(x) \leq R(x) \tag{A.12}$$

where $\mathbf{d}_x \in \partial R(x)$, and where R^{ic} follows Definition 20.

Other smoothing techniques? In order to improve upon the previous results, we tried to look at other smoothing frameworks, hoping that tighter (or easier) bounds may be available to link the smoothed functional F_γ to the original functional F. We do not claim to have look exhaustively for all the possible frameworks, but given what we have found about the following one, we can already draw some conclusions.

Smooth oracles. In [233], the authors introduced the concept of inexact smooth oracle to take into account possible errors when computing the gradient of the smooth approximation. Such errors are none of our concerns, but the oracle proposed can be used in our context. The oracle is defined as follows:

Definition 40. ***Inexact smooth oracle [233].*** *Let R be a convex function on a convex set X. We say that it R equipped with a first order $(\delta, L)-$oracle if for any $y \in X$ we can compute a pair $(f_{\delta,L}(y), g_{\delta,L}(y)) \in \mathbb{R} \times X^*$ such that for all $x \in X$*

$$0 \leq R(x) - (f_{\delta,L}(y) + \langle g_{\delta,L}(y), x - y\rangle) \leq \frac{L}{2}\|x - y\|^2 + \delta \tag{A.13}$$

Remark 25. *A $(\delta, L)-$oracle provides a lower $\delta-$approximation of the function value. Taking $x = y$ in Eq. (A.13)*

$$f_{\delta,L}(y) \leq R(y) \leq f_{\delta,L}(y) + \delta \tag{A.14}$$

Remark 26. *A $(\delta, L)-$oracle provides a $\delta-$subgradient of R at $y \in X$, i.e.*

$$g_{\delta,L}(y) \in \partial_\delta R(y) = \{z \in X^* : R(x) \geq R(y) + \langle z, x - y\rangle - \delta,\ \forall x \in X\}. \tag{A.15}$$

Under the assumption that the function ω used to define an inf-conv γ-approximation of R is such that $\omega(\mathbf{0}) = 0$, an inf-conv approximation constitutes an inexact smooth oracle for R.

Lemma 33. ***Inf-conv smooth approximation are inexact smooth oracles.*** *Consider Definition 20, where $\omega(\mathbf{0}) = 0$. Suppose that R is subdifferentiable over X. For any $x \in X$ the following holds (Lemma 7):*

$$R(x) - \gamma\omega^*(\boldsymbol{d}_x) \leq R_\gamma^{ic}(x) \leq R(x) \tag{A.16}$$

where $\boldsymbol{d}_x \in \partial R(x)$. Then, $(R_\gamma^{ic}, \nabla R_\gamma^{ic})$ is a $\left(\gamma \sup_{x\in X} \sup_{\boldsymbol{d}\in\partial R(x)} \omega^*(\boldsymbol{d}_x), \frac{1}{\sigma\gamma}\right)$ −oracle for R.

Proof. If $\omega(\mathbf{0}) = 0$ one can rewrite Lemma 7 into the following:

$$0 \leq R(x) - R_\gamma^{ic}(x) \leq \mu\omega^*(\mathbf{d}_x). \tag{A.17}$$

Taking the supremum over all x of the right-hand side yields the desired bound on the function values. Now, for the subgradients, using point (b) of [132, Theorem 4.1] which states that R_γ^{ic} has a gradient which is $\frac{1}{\sigma\gamma}$-Lipschitz (recall that $1/\sigma$ is the Lipschitz constant of ω).

Take $\nabla R_\gamma^{ic}(x)$ as the subgradient estimate, and a direct application of the descent lemma (Chapter 2, Lemma 1) yields the desired result: R_μ^{ic} is a first order inexact oracle of R. □

We expect other smoothing framework to be equivalent, under reasonable assumptions, to those already investigated in this manuscript. Therefore, we should probably look in other directions to improve upon our error bound on the decrease of the fine level objective function after a coarse correction.

A.2.2 Beyond FISTA?

In this section, we present a method we tried to better match the evolution of the inertia to the impact of multilevel steps, without losing any convergence guarantees.

In some experiments, we noticed that using the standard update rule for $t_{h,k}$ could be detrimental to the asymptotic convergence speed of our multilevel algorithm. We think the reason is quite intuitive: iteration where we use a coarse model tend to bring us faster closer to the minimum (when compared to the same iteration without coarse correction), which reduces the size of proximal-gradient steps faster than what is expected by the dynamics of FISTA. We can consider increasing the inertia when using coarse corrections to circumvent this problem. The idea is to "boost" the speed of convergence of the sequence $\alpha_{h,k}$ to 1 when $k \to +\infty$.

In the framework of [69, 70], a simple way of doing so is to "saturate" the Nesterov rule : we modify the Chambolle-Dossal sequence slightly so that t_k grows faster when we use coarse iterations while keeping the same convergence guarantees of Theorem 5. For example, we can take the following sequence:

$$\tilde{t}_k = \left(\frac{k + a + b_k - 1}{a}\right)^d \tag{A.18}$$

where the sequence of $(b_k)_{n\in\mathbb{N}}$ is positive and increasing (but not necessarily strictly). As it is crucial for the convergence of the algorithm to guarantee that $\tilde{t}_{k-1}^2 - \tilde{t}_k^2 + \tilde{t}_k \geq 0$, we investigate which type of sequences $(b_k)_{n\in\mathbb{N}}$ could work in the next lemma.

Lemma 34. *The sequence of $(\tilde{t}_k)_{k\in\mathbb{N}}$ defined by (A.18) is such that $\tilde{t}_{k-1}^2 - \tilde{t}_k^2 + \tilde{t}_k$ is positive (resp. strictly positive) if the sequence of $(b_k)_{k\in\mathbb{N}}$ is increasing and such that:*

$$b_k - b_{k-1} \leq \frac{a^d}{2d} - 1 \text{ (resp. } b_k - b_{k-1} < \frac{a^d}{2d} - 1). \tag{A.19}$$

Proof. As in proof of [70, Lemma 3.2] (appendix A.1), we can notice that :

$$(k + a + b_k - 1)^{2d} - (k + a + b_{k-1} - 2)^{2d} = \int_{k+a+b_{k-1}-2}^{n+a+b_k-1} (2d)t^{2d-1}dt$$

Then if $d \in \left[\frac{1}{2}, 1\right]$ by bounding the integral :

$$(k + a + b_k - 1)^{2d} - (k + a + b_{k-1} - 2)^{2d} \leq (2d)(b_k - b_{k-1} + 1)(k + a + b_k - 1)^d$$

and similarly if $d \in \left[0, \frac{1}{2}\right[$,

$$(k+a+b_k-1)^{2d}-(k+a+b_{n-1}-2)^{2d} \leq (2d)(b_k-b_{n-1}+1) \leq (2d)(b_k-b_{k-1}+1)(k+a+b_k-1)^d$$

using the condition [70, Definition 3.1] (Equation (4.27)). Thus :

$$\tilde{t}_{k-1}^2 - \tilde{t}_k^2 + \frac{2d(b_k - b_{k-1} + 1)}{a^{2d}}(k + a + b_k - 1)^d \geq 0$$

And finally :

$$\tilde{\rho}_k = \tilde{t}_{k-1}^2 - \tilde{t}_k^2 + \tilde{t}_k \geq \left(\frac{1}{a^d} - \frac{2d(b_k - b_{k-1} + 1)}{a^{2d}}\right)(k + a + b_k - 1)^d$$

We must finally have :

$$\begin{aligned} 0 &\leq \frac{1}{a^d} - \frac{2d(b_k - b_{k-1} + 1)}{a^{2d}} \\ \Leftrightarrow b_k - b_{k-1} &\leq \frac{a^d}{2d} - 1 \end{aligned}$$

By the condition [70, Definition 3.1], we have: $a > \max(1, (2d)^{\frac{1}{d}})$ which implies $a > (2d)^{\frac{1}{d}}$ and thus $\frac{a^d}{2d} > 1$, which concludes the proof. □

We therefore have after N iterations that $b_N \leq N\left(\frac{a^d}{2d} - 1\right)$. We can saturate the inertia by adding

$$\left(\frac{a^d}{2d} - 1\right)\mu, \tag{A.20}$$

with $\mu \in [0, 1[$ to the sequence each time we use the coarse models. Imposing $\mu < 1$ keeps the strict positivity of $\tilde{\rho}_n$ for all $n \geq 1$.

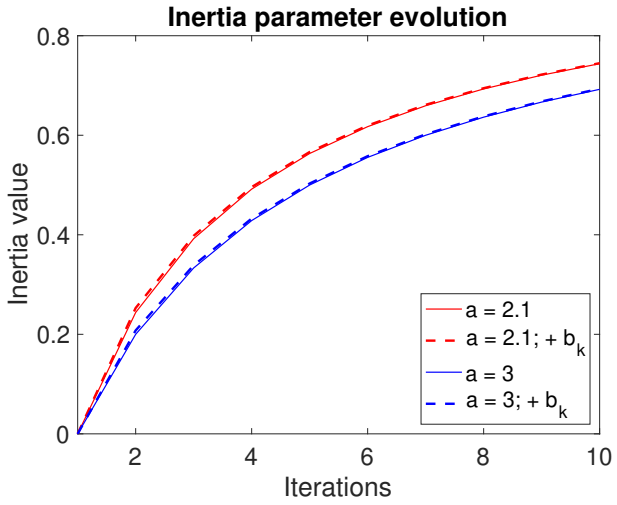

Figure A.1: Comparison of the evolution of the inertia with and without the added b_k. One can easily see that differences between two curves for a given value of a are quite small. Here $\mu = 0.99$ so that $b_0 = 0$ and $b_k = b_{k-1} + (a/2 - 1)\mu$.

Although theoretically promising as we can maintain the same convergence guarantees, in practice the increase of the inertia is difficult to tune. One can only add a small amount on the t_k between two iterates such that it cannot compensate the lack of inertia of our method if only added when coarse correction are used (see a comparison with and without this added inertia in Figure A.1). This raises the question of where to add the missing inertia which we cannot say for now. Could we go beyond Nesterov's rule [75]?

A.2.3 Quick overview of unaddressed hurdles

In this section, we present some known issues about multilevel algorithm in general, that we did not address with the design of IML FISTA.

Null space of the information transfer operators. With multilevel algorithms, one aim to reduce the total computation time to reach a minimizer of the objective function, or an approximate solution. Therefore, in practice, it is almost always chosen to define our coarse models in space of lower dimensions. This leads to information transfer operators having null spaces of large dimension. The problem one can then face is that the projection of the gradient at the fine level to the coarse level may be null.

To circumvent this difficulty it was proposed for instance in [112] to create multiple information transfer operators $(I_h^H)_j$ with $j \in \{1, \dots, P\}$, such that the span of all the rows of the $(I_h^H)_j$ is equal to the whole fine level space. It guarantees the existence of at least one operator $(I_h^H)_j$ such that $\|(I_h^H)_j \nabla F_{h,\gamma_h}\| > 0$.

A cycle browsing between all the operators along the iterations is then used in practice to exploit this property, while preventing the coarse model to be inefficient.

Nevertheless, it is quite involving to create multiple operators, and to check that all the coarse models derived from these operators are useful. Moreover, multilevel algorithms are often more beneficial to the convergence at the beginning of the optimization, where the chances of the coarse correction being in the null space of the information transfer

operator seem to be minimal. We thus choose not to consider such idea in our own experiments.

Constrained optimization problems. In imaging applications, the regularization often contains an indicator function modeling a set of closed convex constraints C (e.g. positiveness of the pixels which is crucial for physical interpretation of the results).

For such convex constrained optimization problems, the definition of convergent multilevel algorithm is not straightforward. In our framework and most of the multilevel frameworks that we know of, we can define a coarse model that will yield a descent direction at the fine level, but we have no guarantee of obtaining feasible descent directions.

And as it is well known that if you replace the gradient descent by a descent direction in the projected gradient descent algorithm, in general the descent property is lost [55], therefore a projection onto the set of constraints of the coarse correction will not work. We need therefore to find a way to guarantee feasibility.

Let us consider that a coarse update has been computed and sent to the fine level. The problem of finding a feasible descent direction can be formulated as:

$$\begin{aligned} &\min_{\alpha>0} F_h\left(x_{h,k} + \alpha I_H^h(x_{H,k,m} - x_{H,k,0})\right) \\ &\text{subject to } x_{h,k} + \alpha I_H^h(x_{H,k,m} - x_{H,k,0}) \in C \end{aligned} \tag{A.21}$$

Sadly there is no guarantee that this problem has a solution $\widehat{\alpha}$ such that:

$$F_h\left(x_{h,k} + \widehat{\alpha} I_H^h(x_{H,k,m} - x_{H,k,0})\right) < F_h(x_{h,k}). \tag{A.22}$$

If no feasible direction is found, taking $\alpha = 0$ at least ensures that this iteration will not be detrimental to the rest of the optimization. As multilevel update involve some computation, this is obviously inefficient.

Even for one of the simplest set of constraints C, positive constraints, there does not exist a simple way to guarantee descent. For the sake of the argument, suppose that we have chosen an information transfer operator I_H^h with only positive coefficients (Equation (3.12) in Chapter 2). This cannot ensure, given a positive $x_{h,k}$ positive and a positive coarse iterate $x_{H,k,m}$ (by construction $x_{H,k,0}$ is positive), that there exists $\alpha > 0$ such that $x_{h,k} + \alpha I_H^h(x_{H,k,m} - x_{H,k,0})$ is also positive. For instance, take $x_{h,k}$ on the boundary of C, and $I_H^h(x_{H,k,m} - x_{H,k,0})$ may point out of the set of constraints.

To get closer to what we want to do, some inspiration could be taken from Frank-Wolfe algorithms (or Conditional Gradient methods) which correspond to the following iterations [234]:

$$v_k \in \arg\min_{v \in C} \langle \nabla f(x_k), v \rangle \tag{A.23}$$

$$x_{k+1} = x_k + \gamma_k(v_k - x_k) \tag{A.24}$$

These methods are commonly used for smooth constrained optimization as projection onto C may not be simple. One can see some connections between the concept of first order coherence and the linearization used in Frank-Wolfe iterations. Sadly we have not been able to link the two together to obtain a multilevel algorithm for constrained optimization. We think it would be an interesting direction to follow, even if Frank-Wolfe algorithm have known flaws [235].

In the smooth case, provided that the set of constraints can be described by a set of linear inequalities, the projection of the gradient to the coarse level can be done with respect to these constraints by modeling them as a manifold upon which gradient descent occurs [108]. Taking into account these constraints greatly improve the speed of reconstruction on tomography applications, but no convergence proof of the algorithm to a minimizer was provided.

The same assumptions that allowed us to obtain convergence of our algorithm in Theorems 4 and 5 (Chapter 4) are fulfilled for the algorithm proposed by [108], hence providing a direct convergence proof for this algorithm.

A.3 Chapter 4 – Extension of IML FISTA's framework to other multilevel algorithms

In this section, we present a strategy to extend the framework of IML FISTA to other multilevel algorithms.

To prove convergence of our algorithm IML FISTA, we only added a small assumption on the iterations: a finite number of coarse corrections may be computed. In practice, this is not a problem at all as we compute a finite number of iterations anyway. This idea being quite simple, we asked ourselves if it could be reused to construct other convergent multilevel algorithms. In the next section, we will present an abstract and simple convergence principle which formalize this.

A.3.1 Abstract convergence principle

The proof of Theorem 4 and 5 rely on the assumption that the underlying fine level algorithm can manage errors whose sum of the norms is finite. Most of the first order algorithms can handle such errors. Formally, let us say that we want to minimize a function f and that we have an oracle $\mathcal{M}_f : \mathcal{H} \mapsto \mathcal{H}$ (whose properties will be presented later) and let $(x_k)_{k\in\mathbb{N}}$ be a sequence in $\mathcal{H}$ constructed by the following recursion:

$$x_{k+1} = \mathcal{M}_f(x_k) + \epsilon_k \tag{A.25}$$

where ϵ_k is a sequence in $\mathcal{H}$.

Assumption 9. *$\mathcal{M}_f$ is such that:*

$$\lim_{k\to+\infty} x_k = \hat{x} \tag{A.26}$$

and that there exists a function $g : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}^+$ such that:

$$\sum_{k\in\mathbb{N}} g(k, \|\epsilon_k\|) < +\infty \tag{A.27}$$

The function g controls how the error behave with respect to the iteration number. In Theorem 5, in the case of type 2 approximations, we had:

$$g(k, \|\epsilon_k\|) = k^{2d}\|\epsilon_k\|$$

We now construct another sequence intertwining $\mathcal{M}_f$ with an **ML** step, which constitutes the basic step of IML FISTA. For some iterations $j \in \mathbb{N}$ we have:

$$x_{j+1} = \mathcal{M}_f(\mathbf{ML}(x_j)) \tag{A.28}$$

This is equivalent to:

$$x_{j+1} = \mathcal{M}_f(x_j) + \Big(\mathcal{M}_f(\mathbf{ML}(x_j)) - \mathcal{M}_f(x_j)\Big) \tag{A.29}$$

The error is here the difference between what would have been computed by $\mathcal{M}_f$ without the coarse correction and the actual value.

$$\|x_{j+1} - \mathcal{M}_f(x_j)\| = \|\mathcal{M}_f(\mathbf{ML}(x_j)) - \mathcal{M}_f(x_j)\| \tag{A.30}$$

We need thus to show that the right hand-side is bounded. For that we make the following assumption:

Assumption 10. *$\mathcal{M}_f$ is M-Lipschitz or M-non expansive with $M > 0$.*

Gradient, proximal operators and other classical operators respect this assumption [60]. This assumption yields:

$$\|\mathcal{M}_f(\mathbf{ML}(x_j)) - \mathcal{M}_f(x_j)\| \leq M\|\mathbf{ML}(x_j) - x_j\| \tag{A.31}$$

Then, $\|\mathbf{ML}(x_j) - x_j\|$ is bounded, which is straightforward if a finite number of iterations at coarse level are employed. Now if we compute a finite number of **ML** steps, it is obvious that for all proper functions g:

$$\sum_{k\in\mathbb{N}} g(k, \mathcal{M}_f(\mathbf{ML}(x_k)) - \mathcal{M}_f(x_k)\|) < +\infty \tag{A.32}$$

Therefore, we have the following convergence result:

Theorem 12. *Suppose that Assumptions 9 and A.3.1 are fulfilled. Let $p \in \mathbb{N}^*$. Suppose that the sequence $(x_k)_{k\in\mathbb{N}}$, is generated by the following algorithm:*

$$x_{k+1} = \begin{cases} \mathcal{M}_f(\mathbf{ML}(x_k)) & p \text{ times} \\ \mathcal{M}_f(x_k) & \text{otherwise} \end{cases} \tag{A.33}$$

Then,

$$\lim_{k\to+\infty} x_k = \hat{x}. \tag{A.34}$$

Proof. The proof is direct from the previous discussion. □

A.3.2 A multilevel primal-dual method

In this section, in order to show that this convergence principle is useful, we will consider the primal-dual algorithm proposed in [140, 143]. It is natural to wonder, given that we deal with non-proximable penalties due to composition with linear operators in Chapter

5, if we could accelerate, with multilevel methods, the convergence of the algorithms that circumvent that inexactness.

This primal-dual algorithm is tailored to solve the following problem[1]:

$$\widehat{x} \in \underset{x \in \mathcal{H}}{\operatorname{Argmin}}\, F(x) := f_1(x) + f_2(x) + f_3(\mathrm{D}x) \tag{A.35}$$

where $\mathcal{H}$, $\mathcal{G}$ are real Hilbert spaces and

- for all i, f_i is proper, convex and lower semi-continuous,
- f_1 is differentiable with β-Lipschitz continuous gradient,
- the proximity operators of f_2 and f_3 are explicitly available or can be estimated,
- and $\mathrm{D} : \mathcal{H} \to \mathcal{G}$ is a bounded linear operator.

Such problem encompasses a wide range of optimization problems, including those we considered in this manuscript. For instance, the proposed algorithm has been used to solve formulation of the radio-interferometric imaging problem we presented in Chapter 6. The algorithm can also be seen as a forward-backward or Douglas-Rachford splitting algorithm depending on the context, thus generalizing them.

To solve this problem, [140, 143] first formulate the dual problem as follows:

$$\widehat{y} \in \underset{y \in \mathcal{G}}{\operatorname{Argmin}} = (f_1 + f_2)^*(-\mathrm{D}^* y) + f_3^*(y) \tag{A.36}$$

The algorithm solves both the primal and the dual problem jointly. Combine for instance these two minimization problem into the search of a saddle point of the Lagrangian:

$$\mathrm{Find}(\hat{x}, \hat{y}) \in \arg\min_{x \in \mathcal{H}} \max_{y \in \mathrm{dom}(f_3^*)} [f_1(x) + f_2(x) - f_3^*(y) + \langle \mathrm{D}x, y \rangle] \tag{A.37}$$

From Karush-Kuhn-Tucker theory, we have that if $(\widehat{x}, \widehat{y}) \in \mathcal{H} \times \mathcal{G}$ is a solution to the monotone variational inclusion

$$\text{Find } (\widehat{x}, \widehat{y}) \in \mathcal{H} \times \mathcal{G} \text{ such that } \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial f_2(\widehat{x}) + \mathrm{D}^* \widehat{y} + \nabla f_1(\widehat{x}) \\ -\mathrm{D}\widehat{x} + \partial f_3^*(\widehat{y}) \end{pmatrix} \tag{A.38}$$

Then $\widehat{x}$ is a solution to Problem (A.35) and $\widehat{y}$ a solution to Problem (A.36). Under simple assumptions [140, 143], solutions to Problems (A.35) and (A.36) provide solutions to Problem (A.38). There exist several ways to organize the Primal-Dual algorithm that will solve Problem (A.38), but for the sake of conciseness we will only consider Algorithm 11 and Algorithm 12.

The proof of convergence of Algorithms 11 and 12 is given in [140, 143].

[1] The notation is different from the one used in the previous section, and in [140, 143], as we face the sum of three functionals (which were named f, g and h in [140]).

Algorithm 11 Condat-Vũ Algorithm 1 [140, 143]

1: $\tau > 0$, $\sigma > 0$, $(\rho_n)_{n\in\mathbb{N}}$,
2: $(x_0, y_0) \in \mathcal{H} \times \mathcal{G}$.
3: **while** Stopping criterion is not met **do**
4: $\quad \tilde{x}_{n+1} := \text{prox}_{\tau f_2}\left(x_n - \tau(\nabla f_1(x_n) + e_{f_1,n}) - \tau \mathrm{D}^* y_n\right) + e_{f_2,n}$,
5: $\quad \tilde{y}_{n+1} := \text{prox}_{\sigma f_3^*}\left(y_n + \sigma \mathrm{D}(2\tilde{x}_{n+1} - x_n)\right) + e_{f_3,n}$,
6: $\quad (x_{n+1}, y_{n+1}) := \rho_n(\tilde{x}_{n+1}, \tilde{y}_{n+1}) + (1-\rho_n)(x_n, y_n)$.
7: **end while**

Algorithm 12 Condat-Vũ Algorithm 2 [140, 143]

1: $\tau > 0$, $\sigma > 0$, $(\rho_n)_{n\in\mathbb{N}}$,
2: $(x_0, y_0) \in \mathcal{H} \times \mathcal{G}$.
3: **while** Stopping criterion is not met **do**
4: $\quad \tilde{y}_{n+1} := \text{prox}_{\sigma f_3^*}\left(y_n + \sigma \mathrm{D}(x_n)\right) + e_{f_3,n}$,
5: $\quad \tilde{x}_{n+1} := \text{prox}_{\tau f_2}\left(x_n - \tau(\nabla f_1(x_n) + e_{f_1,n}) - \tau \mathrm{D}^*(2\tilde{y}_{n+1} - y_n)\right) + e_{f_2,n}$,
6: $\quad (x_{n+1}, y_{n+1}) := \rho_n(\tilde{x}_{n+1}, \tilde{y}_{n+1}) + (1-\rho_n)(x_n, y_n)$.
7: **end while**

Multilevel primal-dual algorithm. Following what we did in the previous section, we want to add a coarse correction before the iteration of the primal-dual algorithm that computes a gradient step on f_1 (cf Chapter 4, Lemma 14).

Thus, we propose to compute a coarse correction on both the primal and dual variables before:

- step 4 in Algorithm 11,
- step 5 in Algorithm 12.

Such coarse correction step would be written as:

$$(\bar{x}_n, \bar{y}_n) := \mathbf{ML}(x_n, y_n). \tag{A.39}$$

We compare the two subsequent algorithms, to highlight their differences, and the difficulty we face in constructing a multilevel primal-dual algorithm with respect to the construction of multilevel forward-backward algorithm. First, we present the iterations of the multilevel primal-dual algorithm 1.

$$(\bar{x}_n, \bar{y}_n) = \mathbf{ML}(x_n, y_n) \tag{A.40}$$
$$\tilde{x}_{n+1} = \text{prox}_{\tau f_2}\left(\bar{x}_n - \tau(\nabla f_1(\bar{x}_n) + e_{f_1,n} + \mathrm{D}^* \bar{y}_n)\right) + e_{f_2,n}, \tag{A.41}$$
$$\tilde{y}_{n+1} = \text{prox}_{\sigma f_3^*}\left(y_n + \sigma \mathrm{D}(2\tilde{x}_{n+1} - x_n)\right) + e_{f_3,n}, \tag{A.42}$$
$$(x_{n+1}, y_{n+1}) = \rho_n(\tilde{x}_{n+1}, \tilde{y}_{n+1}) + (1-\rho_n)(x_n, y_n). \tag{A.43}$$

As we incorporate the "error" created by the multilevel step in $e_{f,1}$, the dual variable $\bar{y}_n$ is only used in Equation (A.41) to obtain $\tilde{x}_{n+1}$, and not in Equation (A.42) to obtain $\tilde{y}_{n+1}$.

The primal-dual algorithm 1 was inherently asymmetric with respect to the primal and dual variables [140], its multilevel version further accentuates this asymmetry.

If we now look at our multilevel primal-dual algorithm 2, we have the following iterations:

$$(\bar{x}_n, \bar{y}_n) = \mathbf{ML}(x_n, y_n) \tag{A.44}$$
$$\tilde{y}_{n+1} = \text{prox}_{\sigma f_3^*}(y_n + \sigma \mathrm{D}(x_n)) + e_{f_3,n}, \tag{A.45}$$
$$\tilde{x}_{n+1} = \text{prox}_{\tau f_2}(\bar{x}_n - \tau(\nabla f_1(\bar{x}_n) + e_{f_1,n} + \mathrm{D}^*(2\tilde{y}_{n+1} - y_n)) + e_{f_2,n}, \tag{A.46}$$
$$(x_{n+1}, y_{n+1}) = \rho_n(\tilde{x}_{n+1}, \tilde{y}_{n+1}) + (1-\rho_n)(x_n, y_n). \tag{A.47}$$

Again, we cannot use the dual variable $\bar{y}_n$ in Equation (A.45) to obtain $\tilde{y}_{n+1}$. We also cannot use $\bar{x}_n$. This version is clearly worse than the first one, as the **ML** step is not at all taken into account in the update of the dual variable, while it is in the first one through $\tilde{x}_{n+1}$ (Equation (A.42)). It remains now to construct such a primal-dual coarse correction.

Multilevel double descent. We will denote as F_H^{primal} and F_H^{dual} the primal and dual objective functions at the coarse level. We will denote by $\mathcal{H}_h$ (resp. $\mathcal{G}_h$) the space of the fine level primal variable (resp. dual variable) and by $\mathcal{H}_H$ (resp. $\mathcal{G}_H$) the space of the coarse primal variable (resp. dual variable). To lighten the notation, we will refer to as I_h^H and I_H^h, both information transfer operators, the space of the variables being implicit. At a pair of primal-dual points (x_h, y_h), the first order coherence on the primal and on the dual problems would be, given the Karush-Kuhn-Tucker conditions of Equation (A.38):

$$\begin{pmatrix} \nabla F_H^{\text{primal}}(x_{H,0}) \\ \nabla F_H^{\text{dual}}(y_{H,0}) \end{pmatrix} := \begin{pmatrix} I_h^H(\nabla^\gamma f_2(x_h) + \mathrm{D}^* y_h + \nabla f_1(x_h)) \\ I_h^H(-\mathrm{D}x_h + \nabla^\mu f_3^*(y_h)) \end{pmatrix} \tag{A.48}$$

The **ML** step would then be:

$$\mathbf{ML}\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} x_n + \bar{\tau}_{x,n} I_H^h(x_{H,n,m} - x_{H,n,0}) \\ y_n + \bar{\tau}_{y,n} I_H^h(y_{H,n,m} - y_{H,n,0}) \end{pmatrix} \tag{A.49}$$

The error term $e_{f_1,n}$ would thus read as:

$$\begin{aligned} e_{f_1,n} =& \nabla f_1(x_n) - \nabla f_1(\bar{x}_n) \\ &+ (\tau)^{-1}\bar{\tau}_{x,n} I_H^h(x_{H,n,m} - x_{H,n,0}) \\ &+ \bar{\tau}_{y,n} D^* I_H^h(y_{H,n,m} - y_{H,n,0}) \end{aligned} \tag{A.50}$$

when a multilevel step is performed and 0 otherwise. Therefore, provided that:

$$\left\| \begin{matrix} \bar{\tau}_{x,n} I_H^h(x_{H,n,m} - x_{H,n,0}) \\ \bar{\tau}_{y,n} \mathrm{D}^* I_H^h(y_{H,n,m} - y_{H,n,0}) \end{matrix} \right\| < +\infty. \tag{A.51}$$

The error term $e_{f,1,n}$ is bounded for all $n \in \mathbb{N}$. If it has a finite number of non-zero terms, we can apply the convergence principle presented in the previous section to the multilevel primal-dual algorithm, and recover the following convergence guarantees (from [140]).

Theorem 13. ***Convergence of Multilevel Primal-Dual Algorithm 1 11 [140, 143].*** *Let $\tau > 0$, $\sigma > 0$ and the sequences $(\rho_n)_{n\in\mathbb{N}}$, $(e_{f_1,n})_{n\in\mathbb{N}}$, $(e_{f_2,n})_{n\in\mathbb{N}}$, $(e_{f_3,n})_{n\in\mathbb{N}}$, be the parameters of Multilevel Primal-Dual Algorithm 1. Suppose that we compute a finite number of **ML** steps $p > 0$. Let β bet the Lipschitz constant of f_1. Suppose that $\beta > 0$ and that the following hold:*

1. $\frac{1}{\tau} - \sigma\|\mathrm{D}\|^2 \geq \frac{\beta}{2}$,
2. $\forall n \in \mathbb{N}$, $\rho_n \in]0, \delta[$, *where we set* $\delta := 2 - \frac{\beta}{2}\left(\frac{1}{\tau} - \sigma\|\mathrm{D}\|^2\right)^{-1} \in [1, 2[$,
3. $\sum_{n\in\mathbb{N}} \rho_n(1 - \rho_n) = +\infty$,
4. $\sum_{n\in\mathbb{N}} \rho_n\|e_{f_1,n}\| < +\infty$ *and* $\sum_{n\in\mathbb{N}} \rho_n\|e_{f_2,n}\| < +\infty$ *and* $\sum_{n\in\mathbb{N}} \rho_n\|e_{f_3,n}\| < +\infty$.

Then there exists a pair $(\widehat{x}, \widehat{y}) \in \mathcal{H} \times \mathcal{G}$ solution to (A.38), such that, in Multilevel Primal-Dual Algorithm 1, the sequences $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ converge weakly to $\widehat{x}$ and $\widehat{y}$, respectively.

A.4 Chapter 7 – Proofs of convergence for algorithm H-BC-FB.

In this section, we detail the proofs of the lemmas, propositions and theorems need to assert the convergence of our H-BC-FB algorithm. First, we recall the following proposition.

Proposition 7. ***Subdifferentiability property [52].*** *Suppose that f in Ψ is continuously differentiable. Then for all $x = (x_1, \ldots, x_L) \in \mathcal{H}_1 \times \ldots \times \mathcal{H}_L$ we have*

$$\partial\Psi(x) = (\nabla_1 f(x) + \partial g_1(x_1), \ldots, \nabla_L f(x) + \partial g_L(x_L)). \tag{A.52}$$

And an extension of Lemma 2 to the non-convex case:

Lemma 35. ***(Non-convex) proximal-gradient descent lemma [64].*** *Let $f : \mathbb{R}^N \mapsto \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant β_f. Let $g : \mathbb{R}^N \mapsto \mathbb{R}$ be a proper, lower semicontinuous function with $\inf_{\mathbb{R}^N} g > -\infty$. If*

$$y \in \mathrm{prox}_{\tau g}(x - \tau\nabla f(x)), \tag{A.53}$$

then for any $0 < \tau < \frac{1}{\beta_f}$

$$f(y) + g(y) + \frac{1}{2}\left(\frac{1}{\tau} - \beta_f\right)\|x - y\|^2 \leq f(x) + g(x). \tag{A.54}$$

Proof. First $\mathrm{prox}_{\tau g}(\cdot)$ is well-defined by [64, Proposition 2]. Thus, for all $x \in \mathbb{R}^N$, there exists $y \in \mathrm{prox}_{\tau g}(x - \tau\nabla f(x))$. This inequality comes directly from [64, Lemma 2], but for completeness of the argument we reproduce it here. By definition of the proximity

operator:

$$y \in \arg\min_{\mathbb{R}^N} \langle z - x, \nabla f(x)\rangle + g(z) + \frac{1}{2\tau}\|z - x\|^2 \tag{A.55}$$

Thus taking $z = x$ we obtain

$$\begin{aligned}\langle y - x, \nabla f(x)\rangle + g(y) + \frac{1}{2\tau}\|y - x\|^2 &\leq \langle x - x, \nabla f(x)\rangle + g(x) + \frac{1}{2\tau}\|x - x\|^2 \\ &\leq g(x)\end{aligned}$$

Now invoking Lemma 1, we have:

$$\langle \nabla f(x), x - y\rangle \leq f(x) - f(y) + \frac{\beta_f}{2}\|x - y\|^2 \tag{A.56}$$

which yields for any $0 < \tau < \frac{1}{\beta_f}$

$$f(y) + g(y) + \frac{1}{2}\left(\frac{1}{\tau} - \beta_f\right)\|x - y\|^2 \leq f(x) + g(x). \tag{A.57}$$

□

Remark 27. *It is interesting to note that the step size of the proximal gradient descent is not controlled by the smooth and potentially non-convex function f but by the convexity of the function g.*

Proof of Proposition 6.

Proof. Indeed, suppose that Assumption 5 A4 is true and let $\mathbf{x}, \mathbf{v} = (v_\ell)_{\ell=1}^L \in \mathcal{H} = \oplus_{\ell=1}^L \mathcal{H}_\ell$. Note that

$$|||\nabla f(\mathbf{x} + \mathbf{v}) - \nabla f(\mathbf{x})|||^2 = \sum_{\ell=1}^L \|\nabla_\ell f(\mathbf{x} + \mathbf{v}) - \nabla_\ell f(\mathbf{x})\|^2. \tag{A.58}$$

We note by $\odot$ the element-wise multiplication. Note that $\varepsilon \odot \mathbf{v} = (\varepsilon_1 v_1, \varepsilon_2 v_2, \ldots, \varepsilon_L v_L)$. Now define, for every $j \in \{0, \ldots, L\}$, $\mathbf{v}_j = \mathbf{0} \in \mathcal{H}$ and $\mathbf{v}_j = (v_1, \ldots, v_j, 0, \ldots, 0)$ if $j > 0$. Note that $\mathbf{v}_L = \mathbf{v}$ and that

$$(\forall j \in \{1, \ldots, L\}) \quad \mathbf{v}_j - \mathbf{v}_{j-1} = (0, \ldots, v_j, \ldots, 0). \tag{A.59}$$

Then, for every $\ell \in \{1, \ldots, L\}$, triangular inequality and A4 imply

$$\begin{aligned}\|\nabla_\ell f(\mathbf{x} + \varepsilon \odot \mathbf{v}) - \nabla_\ell f(\mathbf{x})\| &= \left\| \sum_{j=1}^L (\nabla_\ell f(\mathbf{x} + \varepsilon \odot \mathbf{v}_j) - \nabla_\ell f(\mathbf{x} + \varepsilon \odot \mathbf{v}_{j-1}))\right\| \\ &\leq \sum_{j=1}^L \|\nabla_\ell f(\mathbf{x} + \varepsilon \odot \mathbf{v}_j) - \nabla_\ell f(\mathbf{x} + \varepsilon \odot \mathbf{v}_{j-1})\| \\ &\leq \sum_{j=1}^L \beta_{\ell,j}\|\varepsilon_j v_j\| \end{aligned} \tag{A.60}$$

and, therefore, from Cauchy-Schwarz in $\mathbb{R}^L$,

$$|||\nabla f(\mathbf{x} + \varepsilon \odot \mathbf{v}) - \nabla f(\mathbf{x})|||^2 \leq \sum_{\ell=1}^L \left(\sum_{j=1}^L \beta_{\ell,j}\|\varepsilon_j v_j\|\right)^2 \leq \sum_{\ell,j=1}^L \beta_{\ell,j}^2 |||\varepsilon \odot \mathbf{v}|||^2, \tag{A.61}$$

deducing that $\beta = \sqrt{\sum_{\ell,j=1}^L \varepsilon_j \beta_{\ell,j}^2}$ is the Lipschitz constant of ∇f with respect to the blocks selected by ε. □

Proof of Lemma 21.

Proof. First, for all $\ell \notin I^n$, $x_\ell^{n+1} = x_\ell^n$. For all $\ell \in I^n$, by applying the first order optimality conditions of the proximity operator (2.22)

$$g_\ell(x_\ell^{n+1}) + \frac{1}{2\tau_\ell^n}\|x_\ell^n - x_\ell^{n+1}\|^2 \leq g(x_\ell^n) + \langle \nabla_\ell f(\mathbf{x}^n), x_\ell^n - x_\ell^{n+1}\rangle \tag{A.62}$$

which we can sum up to obtain

$$\sum_{\ell\in I^n}\left(g_\ell(x_\ell^{n+1}) + \frac{1}{2\tau_\ell^n}\|x_\ell^n - x_\ell^{n+1}\|^2\right) \leq \sum_{\ell\in I^n}\Big(g(x_\ell^n) + \langle \nabla_\ell f(\mathbf{x}^n), x_\ell^n - x_\ell^{n+1}\rangle\Big). \tag{A.63}$$

We now invoke A4 from Assumption 5 and by splitting the scalar product along the blocks we get

$$f(\mathbf{x}^n + [x_1^{n+1} - x_1^n, \ldots, x_L^{n+1} - x_L^n]^T) \leq f(\mathbf{x}^n) + \sum_{\ell\in I^n}\langle \nabla_\ell f(\mathbf{x}^n), x_\ell^{n+1} - x_\ell^n\rangle + \frac{\beta_f^n}{2}\|x_\ell^n - x_\ell^{n+1}\|^2 \tag{A.64}$$

Note that $f(\mathbf{x}^n + [x_1^{n+1} - x_1^n, \ldots, x_L^{n+1} - x_L^n]^T) = f(\mathbf{x}^{n+1})$, and thus

$$\sum_{\ell\in I^n}\Big(\langle \nabla_\ell f(\mathbf{x}^n), x_\ell^n - x_\ell^{n+1}\rangle\Big) \leq f(\mathbf{x}^n) - f(\mathbf{x}^{n+1}) + \frac{\beta_f^n}{2}\sum_{\ell\in I^n}\|x_\ell^n - x_\ell^{n+1}\|^2. \tag{A.65}$$

Combining inequalities (A.63) and (A.65) we obtain:

$$\sum_{\ell\in I^n}\left(g_\ell(x_\ell^{n+1}) + \frac{1}{2\tau_\ell^n}\|x_\ell^n - x_\ell^{n+1}\|^2\right) \leq \sum_{\ell\in I^n} g(x_\ell^n) + f(\mathbf{x}^n) - f(\mathbf{x}^{n+1}) + \frac{\beta_f^n}{2}\sum_{\ell\in I^n}\|x_\ell^n - x_\ell^{n+1}\|^2.$$

We add $\sum_{\ell\notin I^n} g_\ell(x_\ell^n)$ to each side of this inequality, and since $\sum_{\ell\notin I^n} g_\ell(x_\ell^n) = \sum_{\ell\notin I^n} g_\ell(x_\ell^{n+1})$, we have

$$\Psi(\mathbf{x}^{n+1}) + \sum_{\ell\in I^n}\frac{1}{2}\left(\frac{1}{\tau_\ell^n} - \beta_f^n\right)\|x_\ell^n - x_\ell^{n+1}\|^2 \leq \Psi(\mathbf{x}^n) \tag{A.66}$$

Bundle K block iterations together to define the sequence $(\bar{\mathbf{x}}^k)_{k\in\mathbb{N}}$. Let $k \in \mathbb{N}$, using inequality (A.66) on all iterations from $n = k \times K$ to $n = (k+1) \times K - 1$ and then summing the resulting inequalities all together, taking into account that $\mathbf{x}^{k\times K} = \bar{\mathbf{x}}^k$ and $\mathbf{x}^{(k+1)\times K} = \bar{\mathbf{x}}^{k+1}$ we have that:

$$\Psi(\bar{\mathbf{x}}^{k+1}) + \sum_{n=k\times K}^{(k+1)\times K-1}\sum_{\ell\in I^n}\frac{1}{2}\left(\frac{1}{\tau_\ell^n} - \beta_f^n\right)\|x_\ell^n - x_\ell^{n+1}\|^2 \leq \Psi(\bar{\mathbf{x}}^k) \tag{A.67}$$

where for all n, $\beta_f^n = \max_{\ell\in I^n}\beta_\ell$. This is the desired result.

Let us now take $n_0 \in \mathbb{N}^*$. Summing up inequality (A.66) from $n = 0$ to $n_0 - 1$, we obtain

$$\begin{aligned}\sum_{n=0}^{n_0-1}\left(\sum_{\ell\in I^n}\frac{1}{2}\left(\frac{1}{\tau_\ell^n} - \beta_f^n\right)\|x_\ell^n - x_\ell^{n+1}\|^2\right) &\leq \Psi(\mathbf{x}^0) - \Psi(\mathbf{x}^{n_0}) \\ &\leq \Psi(\mathbf{x}^0) - \inf\Psi\end{aligned}$$

and then, taking $C := \min_{n=0,\dots,n_0-1} \frac{1}{2}\left(\frac{1}{\tau_\ell^n} - \beta_f^n\right) > 0$ we get

$$\sum_{n=0}^{n_0-1}\left(\sum_{\ell\in I^n}\|x_\ell^n - x_\ell^{n+1}\|^2\right) \leq \frac{1}{C}\left(\Psi(\mathbf{x}^0) - \inf\Psi\right) < +\infty.$$

The limit when N goes to infinity gives us the desired result. □

Proof of Lemma 20.

Proof. Recall that

$$\left|\left|\left|\bar{\mathbf{x}}^{k+1} - \bar{\mathbf{x}}^k\right|\right|\right| = \sqrt{\sum_{\ell=1}^{L}\|\bar{\mathbf{x}}_\ell^{k+1} - \bar{\mathbf{x}}_\ell^k\|^2} \leq \sum_{\ell=1}^{L}\sqrt{\|\bar{\mathbf{x}}_\ell^{k+1} - \bar{\mathbf{x}}_\ell^k\|^2} \leq \sum_{\ell=1}^{L}\|\bar{\mathbf{x}}_\ell^{k+1} - \bar{\mathbf{x}}_\ell^k\|. \tag{A.68}$$

Now for all $\ell \in \{1,\dots,L\}$, a triangular inequality yields

$$\|\bar{\mathbf{x}}_\ell^{k+1} - \bar{\mathbf{x}}_\ell^k\| \leq \sum_{n=k\times K}^{(k+1)\times K-1}\|x_\ell^{n+1} - x_\ell^n\| = \sum_{n=k\times K}^{(k+1)\times K-1}\sum_{\ell\in I^n}\|x_\ell^{n+1} - x_\ell^n\|. \tag{A.69}$$

Thus, summing up for all ℓ, we obtain

$$\left|\left|\left|\bar{\mathbf{x}}^{k+1} - \bar{\mathbf{x}}^k\right|\right|\right| \leq \left(\sum_{n=k\times K}^{(k+1)\times K-1}\sum_{\ell\in I^n}\|x_\ell^{n+1} - x_\ell^n\|\right). \tag{A.70}$$

□

Proof of point (i) and (ii) of Theorem 8.

Proof. • Point (i) has been proven in Lemma 21. Now for the conciseness of the rest of the proof define

$$\rho_k = \min_{k\times K\leq n\leq (k+1)\times K-1}\left(\frac{1}{\tau_\ell^n} - \frac{\beta_f^n}{2}\right) \tag{A.71}$$

By construction $\rho_k \geq 0$. Also set

$$C_k = \sum_{n=k\times K}^{(k+1)\times K-1}\sum_{\ell\in I^n}\|x_\ell^n - x_\ell^{n+1}\|^2. \tag{A.72}$$

• Proof of point (ii). For all $k \in \mathbb{N}$ and for all $\ell \in \{1,\dots,L\}$ there exists an iteration index $n_\ell \in \mathbb{N}$ such that $k \times K \leq n_\ell \leq (k+1)\times K - 1$ and such that block ℓ receives its last update of cycle k at such iteration. We have thus for all $\ell \in \{1,\dots,L\}$ from the optimality conditions of the proximity operator

$$\frac{x_\ell^{n_\ell} - x_\ell^{n_{\ell+1}}}{\tau_\ell^{n_\ell}} - \nabla_\ell f(\mathbf{x}^{n_\ell}) \in \partial g_\ell(x_\ell^{n_{\ell+1}}).$$

We add $\nabla_\ell f(\bar{\mathbf{x}}^{k+1})$ on both sides, so that we get

$$\frac{x_\ell^{n_\ell} - x_\ell^{n_{\ell+1}}}{\tau_\ell^{n_\ell}} - \nabla_\ell f(\mathbf{x}^{n_\ell}) + \nabla_\ell f(\bar{\mathbf{x}}^{k+1}) \in \partial g_\ell(x_\ell^{n_{\ell+1}}) + \nabla_\ell f(\bar{\mathbf{x}}^{k+1}) \\ = \partial_\ell \Psi(\mathbf{x}^{k+1})$$

where the last equality follows from Proposition 7. We can thus construct an element of $\partial\Psi(\mathbf{x}^{k+1})$ by repeating this construction for all ℓ. Now we want to obtain an upper bound on the norm of this element.

Using the fact that ∇f is β_f-Lipschitz continuous we can bound

$$\|\nabla_\ell f(\bar{\mathbf{x}}^{k+1}) - \nabla_\ell f(\mathbf{x}^{n_\ell})\| \leq \beta_f \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^{n_\ell}\right\| \\ \leq \beta_f D_k$$

where the last inequality is deduced from Lemma 20 and

$$D_k = \sum_{n=k\times K}^{(k+1)\times K-1} \sum_{\ell \in I^n} \|x_\ell^n - x_\ell^{n+1}\|$$

The bound is not tight at all, but the proof of convergence does not require it to be. Now define $\tau_k = \min_{k\times K \leq n \leq (k+1)\times K-1} \tau_\ell^n$. Thus, for all $k \in \mathbb{N}$, there exists an element of $\bar{B}^{k+1} \in \partial\Psi(\mathbf{x}^{k+1})$ whose norm is upper bounded by:

$$\left\|\bar{B}^{k+1}\right\| \leq \left(\frac{1}{\tau_k} + \beta_f\right) D_k. \tag{A.73}$$

□

Proof of Lemma 22.

Proof. We have from Lemma 21 and by Lemma 20 that $\left\|\bar{\mathbf{x}}^{k+1} - \bar{\mathbf{x}}^k\right\| \to 0$ as k goes to infinity, and thus point (ii) and point (iii) hold, see [64, Remark 5 & Lemma 5]. Point (iv) is derived from point (i) [64, Lemma 5].
(i) Let $\mathbf{x}^*$ be a limit point of $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$. By Definition 36 there exists a subsequence $\{\bar{\mathbf{x}}^{k_q}\}_{q\in\mathbb{N}}$ such that $\bar{\mathbf{x}}^{k_q} \to \mathbf{x}^*$. Using the assumption that $\forall \ell$, g_ℓ is lower semicontinuous, it follows that for all ℓ:

$$\liminf_{q\to+\infty} g_\ell(x_\ell^{k_q}) \geq g_\ell(x_\ell^*). \tag{A.74}$$

We now need to show that

$$\limsup_{q\to+\infty} g_\ell(x_\ell^{k_q}) \leq g_\ell(x_\ell^*). \tag{A.75}$$

This, combined with the fact that f is continuous, will be then sufficient to obtain that:

$$\lim_{q\to\infty} \Psi(\bar{\mathbf{x}}^{k_q}) = \Psi(\mathbf{x}^*). \tag{A.76}$$

For all $k \in \mathbb{N}$ and for all $\ell \in \{1, \ldots, L\}$ we denote with $k_\ell \in \mathbb{N}$ the iteration index at which block ℓ has received its last update in the k-th cycle (i.e., at which we compute $x_\ell^{k_\ell+1}$ from $\mathbf{x}^{k_\ell}$, thus $k \times K \leq k_\ell \leq (k+1) \times K - 1$). We have for all ℓ:

$$x_\ell^k = \arg\min_{x_\ell \in \mathcal{H}_\ell} \left\{ \langle x_\ell - x_\ell^{k_\ell}, \nabla_\ell f(\mathbf{x}^{k_\ell}) \rangle + \frac{1}{2\tau_\ell} \|x_\ell - x_\ell^{k_\ell}\|^2 + g_\ell(x_\ell) \right\}.$$

Thus, for $x_\ell = x_\ell^*$ it holds

$$\langle x_\ell^k - x_\ell^{k_\ell}, \nabla_\ell f(\mathbf{x}^{k_\ell})\rangle + \frac{1}{2\tau_\ell}\|x_\ell^k - x_\ell^{k_\ell}\|^2 + g_\ell(x_\ell^k) \leq \langle x_\ell^* - x_\ell^{k_\ell}, \nabla_\ell f(\mathbf{x}^{k_\ell})\rangle + \frac{1}{2\tau_\ell}\|x_\ell^* - x_\ell^{k_\ell}\|^2 + g_\ell(x_\ell^*)$$

The index k_ℓ depends implicitly on k. For the rest of the proof we need to extract the converging subsequence, and to note the dependence to q we will write $k_{q,\ell}$ to indicate the last update received by block ℓ at cycle k_q. Taking then $k = k_q$, we get:

$$\begin{aligned}&\langle x_\ell^{k_q} - x_\ell^{k_{q,\ell}}, \nabla_\ell f(\mathbf{x}^{k_{q,\ell}})\rangle + \frac{1}{2\tau_\ell}\|x_\ell^{k_q} - x_\ell^{k_{q,\ell}}\|^2 + g_\ell(x_\ell^{k_q}) \\ &\leq \langle x_\ell^* - x_\ell^{k_{q,\ell}}, \nabla_\ell f(\mathbf{x}^{k_{q,\ell}})\rangle + \frac{1}{2\tau_\ell}\|x_\ell^* - x_\ell^{k_{q,\ell}}\|^2 + g_\ell(x_\ell^*)\end{aligned} \tag{A.77}$$

Now we look at the limit when q goes to infinity. Using the following properties:

- $\|x_\ell^{k_q} - x_\ell^{k_{q,\ell}}\|$ goes to 0 as q goes to infinity,
- $\nabla_\ell f$ is Lipschitz continuous and the sequence $(\mathbf{x}^n)_{n\in\mathbb{N}}$ is bounded,
- $\|x_\ell^* - x_\ell^{k_{q,\ell}}\| \leq \|x_\ell^* - x_\ell^{k_q}\| + \|x_\ell^{k_q} - x_\ell^{k_{q,\ell}}\|$ and both terms on the right-hand side of the inequality go to 0 as q goes to infinity.

We can deduce that

$$\limsup_{q\to+\infty} g_\ell(x_\ell^{k_q}) \leq g_\ell(x_\ell^*)$$

Now, we know from point (ii) that $\bar{B}^k \to 0$ as $k \to +\infty$. The closedness property of $\partial\Psi$ implies that $0 \in \partial\Psi(\mathbf{x}^*)$, and therefore that $\mathbf{x}^*$ is a critical point of Ψ. □

Proof of point (iii) and (iv) of Theorem 9.

Proof. • Proof of point (iii). Now we follow the path of [64]. Since the sequence $(\bar{\mathbf{x}}^k)_{k\in\mathbb{N}}$ is bounded, there exists a sub-sequence that converges to $\mathbf{x}^*$.

1. As $\{\Psi(\bar{\mathbf{x}}^k)\}_{k\in\mathbb{N}}$ is a non-increasing sequence, and as the limit points set $\mathrm{lp}(\mathbf{x}^0)$ is such that $\lim_{k\to\infty}\mathrm{dist}(\bar{\mathbf{x}}^k, \mathrm{lp}(\mathbf{x}^0)) = 0$ ((ii) of Lemma 22), there exist $k_0 \in \mathbb{N}, \varepsilon > 0, \eta > 0$ such that for all $k > k_0$, $\bar{\mathbf{x}}^k$ belongs to:

$$\Big\{\mathbf{x} \in \mathbb{R}^d : \mathrm{dist}(\mathbf{x}, \mathrm{lp}(\mathbf{x}^0) < \varepsilon\Big\} \cap [\Psi(\mathbf{x}^*) < \Psi(\mathbf{x}) < \Psi(\mathbf{x}^*) + \eta].$$

2. Using now that $\mathrm{lp}(\mathbf{x}^0)$ is nonempty and compact, and that Ψ is constant on it (Lemma 22 (ii) and (iv)), one can apply Lemma 19 so that for any $k > k_0$:

$$\varphi'(\Psi(\mathbf{x}) - \Psi(\mathbf{x}^*))\mathrm{dist}(0, \partial\Psi(\mathbf{x})) \geq 1.$$

3. One can now use point (ii) to upper bound $\mathrm{dist}(0, \partial\Psi(\bar{\mathbf{x}}^k))$: at least one element of $\partial\Psi(\bar{\mathbf{x}}^k)$ has its norm bounded, thus $\mathrm{dist}(0, \partial\Psi(\bar{\mathbf{x}}^k))$ is necessarily equal to or less than this bound. Now define $\rho_1 = \min_k \rho_k$ and $\rho_2 = \max_k \left(\frac{1}{\tau_k} + \beta_f\right)$. We have

$$\mathrm{dist}(0, \partial\Psi(\bar{\mathbf{x}}^k)) \leq \rho_2 D_{k-1} \tag{A.78}$$

This bound then yields:

$$\begin{aligned} &\varphi'(\Psi(\bar{\mathbf{x}}^k) - \Psi(\mathbf{x}^*))\rho_2 D_{k-1} \geq 1, \\ \Longrightarrow\ &\varphi'(\Psi(\bar{\mathbf{x}}^k) - \Psi(\mathbf{x}^*)) \geq \rho_2^{-1} D_{k-1}^{-1}. \end{aligned}$$

4. The concavity of φ yields that:

$$\varphi(\Psi(\bar{\mathbf{x}}^k) - \Psi(\mathbf{x}^*)) - \varphi(\Psi(\mathbf{x}^{k+1}) - \Psi(\mathbf{x}^*)) \geq \varphi'(\Psi(\bar{\mathbf{x}}^k) - \Psi(\mathbf{x}^*)) \left(\Psi(\bar{\mathbf{x}}^k) - \Psi(\mathbf{x}^{k+1})\right). \tag{A.79}$$

5. Now recall that:

$$\rho_1 C_k \leq \Psi(\bar{\mathbf{x}}^k) - \Psi(\mathbf{x}^{k+1})$$

And define the following quantity for all $p, q \in \mathbb{N}$:

$$\Delta_{p,q} := \varphi(\Psi(\mathbf{x}^p) - \Psi(\mathbf{x}^*)) - \varphi(\Psi(\mathbf{x}^q) - \Psi(\mathbf{x}^*))$$

Setting $\rho := \rho_1 \rho_2^{-1} > 0$ we now get:

$$\Delta_{k,k+1} \geq \frac{C_k}{\rho\ D_{k-1}} \tag{A.80}$$

and then:

$$C_k \leq \rho\ \Delta_{k,k+1} D_{k-1}$$

6. First, to simplify the computations we rewrite the expression of both C_k and D_k so that:

$$D_k = \sum_{j=1}^{m_k} a_{k,j} \tag{A.81}$$

where $m_k = \sum_{n=k\times K}^{(k+1)} \mathrm{card}(I^n)$ and $a_{k,j} = \|x_\ell^n - x_\ell^{n+1}\|$ where j browses through n and ℓ by increasing order, meaning $a_{k,1} = \|x_3^{k\times K} - x_3^{k\times K+1}\|$ if at the first iteration of the cycle, block 3 has been updated but not block 1 and 2. Then:

$$C_k = \sum_{j=1}^{m_k} a_{k,j}^2 \tag{A.82}$$

We recall that the 1-trick of the Cauchy inequality implies that:

$$\begin{aligned} &\sum_{j=1}^{m_k} a_{k,j} \leq \sqrt{m_k}\sqrt{\sum_{j=1}^{m_k} a_{k,j}^2} \\ \Leftrightarrow\ &D_k \leq \sqrt{m_k}\sqrt{C_k} \end{aligned}$$

Using $2\sqrt{ab} \leq a + b$ for all $a, b \geq 0$, and writing $M = \max_k m_k$ we get that

$$2D_k \leq M\rho\Delta_{k,k+1} + D_{k-1} \tag{A.83}$$

We now want to sum up this inequality from $i = k_0 + 1$ to a given $k > k_0$ to demonstrate that the sum of left-hand side is bounded and thus converges. It is straightforward to see that:

$$\begin{aligned} 2\sum_{i=k_0+1}^{k} D_i &\leq \sum_{i=k_0+1}^{k} D_{i-1} + M\rho \sum_{i=k_0+1}^{k} \Delta_{i,i+1} \\ &= \sum_{i=k_0}^{k-1} D_i + M\rho \sum_{i=k_0+1}^{k} \Delta_{i,i+1} \\ &\leq \sum_{i=k_0+1}^{k} D_i + D_{k_0} + M\rho \sum_{i=k_0+1}^{k} \Delta_{i,i+1} \\ \implies \sum_{i=k_0+1}^{k} D_i &\leq D_{k_0} + M\rho \sum_{i=k_0+1}^{k} \Delta_{i,i+1} \\ \implies \sum_{i=k_0+1}^{k} D_i &\leq D_{k_0} + M\rho\Delta_{k_0+1,k+1} \end{aligned} \tag{A.84}$$

The last line coming from the fact $\Delta_{p,q} + \Delta_{q,r} = \Delta_{p,r}$ for all $p, q, r \in \mathbb{N}$ [64, Proof of theorem 3.1]. As $\varphi \geq 0$, we have that:

$$\Delta_{k_0+1,k+1} = \varphi(\Psi(\bar{\mathbf{x}}^{k_0+1}) - \Psi(\bar{\mathbf{x}})) - \varphi(\Psi(\bar{\mathbf{x}}^{k+1}) - \Psi(\bar{\mathbf{x}})) \leq \varphi(\Psi(\mathbf{x}^{k_0+1}) - \Psi(\bar{\mathbf{x}}))$$

Then as $D_k \geq \left|\left|\left|\bar{\mathbf{x}}^k - \bar{\mathbf{x}}^{k+1}\right|\right|\right|$ (Lemma 20), this allows us then to conclude then that $\{\bar{\mathbf{x}}^k\}_{k\in\mathbb{N}}$ has finite length:

$$\sum_{k=1}^{\infty} \left|\left|\left|\bar{\mathbf{x}}^k - \bar{\mathbf{x}}^{k+1}\right|\right|\right| < \infty \tag{A.85}$$

• Proof of point (iv). The finite length of the sequence implies that it is a Cauchy sequence and hence a convergent sequence [64, Proof of Theorem 3.1(ii)]. □

Proof of Lemma 23.

Proof. Due to the convexity of g_ℓ for all ℓ, the associated proximity operator is uniquely valued.

For all $\ell \in I^n$, the first order optimality conditions of the proximity operator (2.22) yield:

$$g_\ell(x_\ell^{n+1}) + \frac{1}{\tau_\ell^n}\|x_\ell^n - x_\ell^{n+1}\|^2 \leq g(x_\ell^n) + \langle \nabla_\ell f(\mathbf{x}^n), x_\ell^n - x_\ell^{n+1}\rangle \tag{A.86}$$

The subtle difference with the non-convex case is the factor dividing $\|x_\ell^n - x_\ell^{n+1}\|^2$. With the same derivation as in the non-convex case (proof of Lemma 21), we obtain finally that:

$$\Psi(\bar{\mathbf{x}}^{k+1}) + \left(\sum_{n=k\times K}^{(k+1)\times K-1} \sum_{\ell\in I^n} \left(\frac{1}{\tau_\ell^n} - \frac{\beta_f^n}{2}\right)\|x_\ell^n - x_\ell^{n+1}\|^2\right) \leq \Psi(\bar{\mathbf{x}}^k). \tag{A.87}$$

The summability of the sequences is derived similarly. □

A.5 Chapter 7 – Supplementary literature on optimization with wavelets

In this section, we present some works that used wavelet transform to reduce the dimension of the problem at hand, and solved it by considering the subspaces of the wavelet transform. The idea, at first glance, looks similar to ours, but there are notable differences.

Subspace correction methods. In a couple of articles [236,237], the authors introduce a subspace correction method for problem of the form:

$$\min_{x\in\mathcal{H}} F(x) := \|\mathrm{A}x - z\|^2 + 2\lambda g(x) \tag{A.88}$$

where $\mathrm{A} \in \mathcal{L}(\mathcal{H})$ is a bounded linear operator, $\lambda > 0$ and g (total variation for instance) is a semi-norm for a suitable subspace $\mathcal{H}^g$ of $\mathcal{H}$. The main idea of the two articles is to decompose the variable x into two spaces $\mathcal{V}_1, \mathcal{V}_2$ such $x^0 = u_1^0 + u_2^0 \in \mathcal{V}_1 \oplus \mathcal{V}_2$, like in our BC example tackling wavelet deblurring (Chapter 7, Section 7.2) and iterate:

$$\begin{cases} u_1^{k+1} & \approx & \arg\min_{u_1\in\mathcal{V}_1} F(u_1 + u_2^k), \\ u_2^{k+1} & \approx & \arg\min_{u_2\in\mathcal{V}_2} F(u_1^k + u_2) \\ x^{k+1} & = & u_1^{k+1} + u_2^{k+1} \end{cases} \tag{A.89}$$

Such methods were also employed in [238] for deblurring (or deconvolution) problems using Haar wavelets, where cyclic updates across different resolution levels were combined with the preconditioning effect of subband-specific parameters [237]. The spectral localization properties of wavelets are noted as being suitable for preconditioning [238], partly compensating for the poor conditioning of the inverse problem. The algorithm presented is comparable but differs slightly from a non-linear block Gauss-Seidel method, as the variable u_2^{k+1} is not updated using u_1^{k+1}. Additionally, a significant distinction from block-coordinate methods is that the partial sub-problems in this algorithm may not be solved exactly, whereas block-coordinate methods typically require exact minimization of each (proximal in our case) sub-problems.

Finally, the authors of [236,237] do not prove that their algorithm converges to a minimizer of the functional J in general, even when the algorithm halts (i.e., it can converge to an incorrect solution). In fact, they provide a counterexample demonstrating failure to converge to a minimizer when using a Haar wavelet decomposition [237, Proposition 4.2]

Furthermore, as mentioned in [237], numerical experiments reveal that the beneficial effects of preconditioning are not significantly improved by considering multiple decompositions. It is also observed that consistently using coarse models may be advantageous, as they tend to converge in a similar number of iterations as single-level methods. However, for large images, the single-level method significantly outperforms the decomposition method in terms of CPU time, unlike multilevel methods. Since multilevel approaches only aim for approximate convergence within each subdomain (as described in the algorithm), this suggests that minimizing coarse models to full convergence may not be an optimal strategy.

A.6 Chapter 7 – Proofs of equivalence between multilevel and block-coordinate methods

Proof of lemma 27.

Proof. We reason by induction. We have $\mathrm{A}_J = \mathrm{A}$ and $\mathrm{A}_{J-1} = \Pi_{V,-1}\mathrm{A}(\Pi_{V,-1})^*$. Hence, the statement holds for $\ell = J-1$.

Now suppose that for $\ell \in \{J-L+1, \dots, -2\}$:

$$\mathrm{A}_\ell = \left(\prod_{i=0}^{J-(\ell+1)} \Pi_{V,\ell+i}\right) \mathrm{A} \left(\prod_{i=1}^{J-\ell} \Pi^*_{V,J-i}\right) \tag{A.90}$$

Computing $\mathrm{A}_{\ell-1}$ yields:

$$\begin{aligned}
\mathrm{A}_{\ell-1} &= \Pi_{V,\ell-1}\mathrm{A}_\ell\Pi^*_{V,\ell-1} && \text{(A.91)}\\
&= \Pi_{V,\ell-1}\left(\prod_{i=0}^{J-(\ell+1)} \Pi_{V,\ell+i}\right) \mathrm{A} \left(\prod_{i=1}^{J-\ell} \Pi^*_{V,J-i}\right) \Pi^*_{V,\ell-1} && \text{(A.92)}\\
&= \left(\prod_{i=-1}^{J-(\ell+1)} \Pi_{V,\ell+i}\right) \mathrm{A} \left(\prod_{i=1}^{J-\ell+1} \Pi^*_{V,J-i}\right) && \text{(A.93)}\\
&= \left(\prod_{i=0}^{J-\ell} \Pi_{V,\ell-1+i}\right) \mathrm{A} \left(\prod_{i=1}^{J-\ell+1} \Pi^*_{V,J-i}\right) && \text{(A.94)}
\end{aligned}$$

which concludes the proof for A_ℓ. Projecting an observation vector z to a coarse level is quite straightforward, therefore we won't detail it here. □

Proof of Lemma 28.

Proof. First, we have that for all $\ell \in \{J-L+1, \dots, J-1\}$, the projection of a point $a_{\ell+1} = [a_{J-L}, d_{J-L}, d_{J-L+1}, \dots, d_\ell]$ to the space V_ℓ is given by:

$$a_\ell = \Pi_{V,\ell} a_{\ell+1}. \tag{A.95}$$

Recall that the first order coherence between two levels $\ell, \ell+1 \in \{J-L, J-1\}$ is enforced by the following relationship at a point $a_{\ell+1}$:

$$\begin{aligned}
v_\ell = \Pi_{V,\ell} &\left(\nabla\Big(f_{\ell+1}(\mathrm{A}_{\ell+1}a_{\ell+1} - z_{\ell+1})\Big) + \sum_{i=J-L}^{\ell} \nabla g_{\mu,i}(d_i) + v_{\ell+1}\right)\\
&- \nabla\Big(f_\ell(\mathrm{A}_\ell a_\ell - z_\ell)\Big) - \sum_{i=J-L}^{\ell-1} \nabla g_{\mu,i}(d_i) \qquad \text{(A.96)}
\end{aligned}$$

We now look more closely at the projection of the gradient of level $\ell+1$ to level ℓ. We first have that:

$$\begin{aligned}
\Pi_{V,\ell}\Big(\sum_{i=J-L}^{\ell} \nabla g_{\mu,i}(d_i)\Big) &= \sum_{i=J-L}^{\ell} \Pi_{V,\ell}\nabla g_{\mu,i}(d_i)\\
&= \sum_{i=J-L}^{\ell-1} \nabla g_{\mu,i}(d_i) \qquad \text{(A.97)}
\end{aligned}$$

using that $d_\ell \in W_\ell$ and therefore $\Pi_{V,\ell} d_\ell = 0$. Similarly, we have that:

$$\Pi_{V,\ell} \left(\nabla f_{\ell+1}(\mathrm{A}_{\ell+1} a_{\ell+1} - z_{\ell+1})\right) = \Pi_{V,\ell} \Big(\mathrm{A}^*_{\ell+1} \left(\mathrm{A}_{\ell+1} a_{\ell+1} - z_{\ell+1}\right)\Big) \tag{A.98}$$

and that:

$$\begin{aligned} \nabla f_\ell(\mathrm{A}_\ell a_\ell - z_\ell) &= \Big(\mathrm{A}^*_\ell \left(\mathrm{A}_\ell a_\ell - z_\ell\right)\Big) && \text{(A.99)} \\ &= \Pi_{V,\ell} \mathrm{A}^*_{\ell+1} \Big(\mathrm{A}_{\ell+1} \Pi^*_{V,\ell} a_\ell - \Pi^*_{V,\ell} z_\ell\Big) && \text{(A.100)} \end{aligned}$$

Now recall that $a_{\ell+1} - \Pi^*_{V,\ell} a_\ell = d_\ell = \Pi_{W,\ell} a_{\ell+1}$ and that $z_{\ell+1} - \Pi^*_{V,\ell} z_\ell = \Pi^*_{W,\ell+1} \Pi_{W,\ell} z_{\ell+1}$. Hence,

$$v_\ell = \Pi_{V,\ell} \mathrm{A}^*_{\ell+1} \Big(\mathrm{A}_{\ell+1} \Pi^*_{W,\ell} \Pi_{W,\ell} (a_{\ell+1} - z_{\ell+1})\Big) + \Pi_{V,\ell} v_{\ell+1} \tag{A.101}$$

□

Proof of Lemma 29.

Proof. We reason by first expressing in full v_ℓ without involving $v_{\ell+1}$. The base case is given by Lemma 28. We have that:

$$\begin{aligned} v_\ell &= \Pi_{V,\ell} \mathrm{A}^*_{\ell+1} \Big(\mathrm{A}_{\ell+1} \Pi^*_{W,\ell} \Pi_{W,\ell} (a_{\ell+1} - z_{\ell+1})\Big) && \text{(A.102)} \\ &\quad + \Pi_{V,\ell+1} \mathrm{A}^*_{\ell+2} \Big(\mathrm{A}_{\ell+2} \Pi^*_{W,\ell+1} \Pi_{W,\ell+1} (a_{\ell+2} - z_{\ell+2})\Big) && \text{(A.103)} \\ &\quad + \Pi_{V,\ell} \Pi_{V,\ell+1} v_{\ell+2}. && \text{(A.104)} \end{aligned}$$

From which we can deduce the more general formula:

$$v_\ell = \sum_{i=1}^{J-\ell} \left(\prod_{j=0}^{i-1} \Pi_{V,\ell+j}\right) \mathrm{A}^*_{\ell+i} \Big(\mathrm{A}_{\ell+i} \Pi^*_{W,\ell+i-1} \Pi_{W,\ell+i-1} \left(a_{\ell+i} - z_{\ell+i}\right)\Big) \tag{A.105}$$

It remains now to insert into this equation the explicit form of the operators $\mathrm{A}_{\ell+i}$ and $z_{\ell+i}$. Recall that:

$$\mathrm{A}_{\ell+i} = \left(\prod_{k=0}^{J-(\ell+i+1)} \Pi_{V,\ell+i+k}\right) \mathrm{A} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k}\right)$$

We will split the contribution of the wavelet coefficients $a_{\ell+i}$ and $z_{\ell+i}$ in the following way:

$$\begin{aligned} v_\ell = &\sum_{i=1}^{J-\ell} \left(\prod_{j=0}^{i-1} \Pi_{V,\ell+j}\right) \mathrm{A}^*_{\ell+i} \Big(\mathrm{A}_{\ell+i} \Pi^*_{W,\ell+i-1} d_{\ell+i-1}\Big) \\ &- \sum_{i=1}^{J-\ell} \left(\prod_{j=0}^{i-1} \Pi_{V,\ell+j}\right) \mathrm{A}^*_{\ell+i} \Big(\mathrm{A}_{\ell+i} \Pi^*_{W,\ell+i-1} \Pi_{W,\ell+i-1} z_{\ell+i}\Big) \end{aligned}$$

which yields:

$$\begin{aligned} v_\ell = &\sum_{i=1}^{J-\ell} \left(\prod_{j=0}^{i-1} \Pi_{V,\ell+j}\right) \left(\prod_{k=0}^{J-(\ell+i+1)} \Pi_{V,\ell+i+k}\right) \mathrm{A}^* \left(\mathrm{A} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k}\right) \Pi^*_{W,\ell+i-1} d_{\ell+i-1}\right) \\ &- \sum_{i=1}^{J-\ell} \left(\prod_{j=0}^{i-1} \Pi_{V,\ell+j}\right) \left(\prod_{k=0}^{J-(\ell+i+1)} \Pi_{V,\ell+i+k}\right) \mathrm{A}^* \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k}\right) \left(\Pi^*_{W,\ell+i-1} \Pi_{W,\ell+i-1} z_{\ell+i}\right), \end{aligned}$$

that can be "simplified" to:

$$v_\ell = \sum_{i=1}^{J-\ell} \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \left(\mathrm{A} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k} \right) \Pi^*_{W,\ell+i-1} d_{\ell+i-1} \right) - \sum_{i=1}^{J-\ell} \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k} \right) \left(\Pi^*_{W,\ell+i-1} \Pi_{W,\ell+i-1} \left(\prod_{k=0}^{J-\ell-i} \Pi_{V,\ell+i+k} \right) z \right).$$

We can pass the sum inside the product to obtain:

$$v_\ell = \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^*\mathrm{A} \left(\sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k} \right) \Pi^*_{W,\ell+i-1} d_{\ell+i-1} \right) - \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k} \Pi^*_{W,\ell+i-1} \Pi_{W,\ell+i-1} \left(\prod_{k=0}^{J-\ell-i} \Pi_{V,\ell+i+k} \right) z \right).$$

The term

$$\sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k} \right) \Pi^*_{W,\ell+i-1} d_{\ell+i-1}$$

contains all the contribution of coefficients $[d_\ell, \ldots, d_{J-1}]$ to the correction term v_ℓ. Therefore, we will write in the following that:

$$v_\ell = \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^*\mathrm{A} \left([d_\ell, \ldots, d_{J-1}] \right) - \left(\prod_{k=0}^{J-\ell} \Pi_{V,\ell+k} \right) \mathrm{A}^* \sum_{i=1}^{J-\ell} \left(\prod_{k=1}^{J-\ell-i} \Pi^*_{V,J-k} \Pi^*_{W,\ell+i-1} \Pi_{W,\ell+i-1} \left(\prod_{k=0}^{J-\ell-i} \Pi_{V,\ell+i+k} \right) z \right),$$

so that the proof is simpler to read and follow. □

Résumé long de la thèse en français

Nous présentons ici une traduction de l'introduction et de la conclusion ainsi qu'un résumé détaillé des chapitres de la thèse.

Contexte et motivation

Le domaine des problèmes inverses se réfère à la reconstruction des informations manquantes d'une image ou d'un signal, à partir d'observations partielles ou dégradées de ceux-ci. Dans le cadre classique de la restauration d'images, l'information manquante est l'image originale, mais peut aussi inclure des paramètres décrivant le modèle de dégradation, comme le niveau de bruit, ou le flou.

Un exemple célèbre de problèmes inverses, et probablement l'un des plus anciens, est la découverte par Le Verrier de la planète Neptune en 1846, qui a remarqué que le mouvement d'Uranus ne correspondait pas à celui prédit en ne tenant compte que de l'attraction gravitationnelle de Jupiter et de Saturne. Le Verrier en déduisit alors l'existence d'une autre planète, Neptune, dont l'attraction gravitationnelle pourrait expliquer l'écart entre ses relevés et les prédictions théoriques. Il présenta ses résultats à l'Académie des Sciences le 31 août 1846 et la planète fût observée pour la première fois le 23 septembre 1846 par l'astronome Johann Galle sur la base des prédictions faites par Le Verrier.

L'étude des problèmes inverses dans un cadre formel apparaît au début du 20e siècle et est abordée pour la première fois par Tikhonov en 1943 [1]. L'application de ce cadre à l'imagerie a suivi dans les années 1970 [2] et a toujours été une question centrale de recherche depuis.

Les problèmes de reconstruction d'images sont un exemple de problèmes inverses. Dès qu'une image est acquise par un instrument, qu'il s'agisse d'un appareil photo ou d'un télescope, l'image résultante sera floutée et bruitée. Cependant, dans de nombreux cas, nous comprenons assez bien comment les instruments capturent et dégradent les images. La question se pose alors de savoir comment supprimer cette dégradation, c'est-à-dire comment inverser ce processus.

La dégradation de l'image induit une perte d'information et, par conséquent, aucune reconstruction ne peut être parfaite. Cela a stimulé le développement de méthodes visant à atténuer au mieux les effets de cette perte, d'abord en élaborant des représentations théoriques des images [4, 5], puis en construisant des algorithmes capables de tirer parti de la connaissance de la dégradation et de ces représentations [2,6] pour restaurer l'image. Les développements dans les deux domaines sont encore en cours, aucune représentation

n'étant pleinement satisfaisante pour modéliser les images naturelles (le terme naturel est couramment utilisé pour se référer à ce à quoi une image devrait ressembler) et aucun algorithme d'optimisation n'étant a priori meilleur qu'un autre dans tous les contextes. Dans cette thèse nous nous intéressons plutôt à cette dernière question et avons essayé d'apporter des réponses à la question suivante : quelles sont de bonnes stratégies pour concevoir des algorithmes efficaces capable de restaurer/reconstruire des images de grande taille.

Les principaux défis en optimisation : convergence et passage à l'échelle

Les méthodes de résolution des problèmes inverses sont souvent basées sur des algorithmes d'optimisation dont le but est de minimiser (ou de maximiser, ces deux notions étant équivalentes) une fonction objectif. Cette fonction est construite à partir du problème inverse en question. La construction la plus courante consiste à additionner deux différents termes. Le premier contrôle la proximité d'une image par rapport à l'image observée, compte tenu du vecteur de dégradation (que nous supposons connue) : nous prenons une image, la dégradons, puis la comparons à l'image observée. Ce terme est appelé *attache aux données*. Il garantit que la reconstruction corresponde aux observations. Le second terme contrôle la proximité de l'image avec ce que nous pensons être l'image originale (c'est-à-dire le degré de naturalité de cette image). Ce terme est appelé *régularisation*. Il prend en compte nos *a prioris* sur la reconstruction.

La solution du problème d'optimisation défini par la somme de ces deux termes doit donc être un compromis entre être fidèle aux observations et être fidèle à ces *a prioris*. Notre objectif est de parvenir à un compromis qui conduise à une reconstruction satisfaisante. Par conséquent, le choix d'un algorithme d'optimisation capable de trouver la solution du problème, qui correspond donc à ce compromis, est crucial.

Dès lors, l'une des questions les plus importantes lors de la conception d'un algorithme d'optimisation est de savoir s'il peut garantir que la solution produite est la solution optimale du problème, c'est-à-dire va-t-il atteindre la véritable solution du problème ?

Une autre question importante est celle du coût en temps de calcul de l'algorithme. La dimension des problèmes d'optimisation considérés de nos jours est souvent très élevée, allant du million (pour les problèmes classiques d'imagerie couleur) au milliard de variables (pour les problèmes d'imagerie hyperspectrale). Cela crée un goulot d'étranglement computationnel, en sus de celui du stockage de ces variables : chaque itération d'un algorithme d'optimisation est coûteuse, et nous voulons donc réduire autant que possible le nombre d'itérations pour atteindre une solution optimale, c'est-à-dire augmenter la vitesse de convergence de l'algorithme.

Les approches multiniveaux fournissent un moyen de réduire le coût, en temps de calcul, pour atteindre une solution de notre problème, en modifiant certaines itérations. En pratique, elles permettent d'améliorer considérablement la vitesse de convergence des algorithmes.

Approches multiniveaux : une intuition

Pour décrire correctement les motivations qui sous-tendent l'optimisation multiniveaux, je préfère commencer par une analogie[2] plutôt que par un argument technique. Je m'attends à ce que tout lecteur ayant travaillé suffisamment longtemps en optimisation, ait pensé à celle-ci au moins une fois en des termes similaires.

Imaginez que vous ayez les yeux bandés au sommet d'une montagne. Vous voulez vous rendre en bas de la vallée le plus rapidement possible. En explorant à tâtons autour de vous, vous pouvez déduire la pente de la montagne et prendre ainsi une direction de descente. Certaines directions sont meilleures que d'autres, et vous pouvez trouver ce que nous appelons la direction de descente la plus forte (c'est-à-dire la direction indiquée par le gradient de la pente), qui maximisera votre vitesse de descente, un pas à la fois. Cependant, chacun de vos pas ne peut pas aller très loin. Vous devez encore tâtonner à chaque pas pour trouver la direction de descente la plus forte.

Une solution classique à ce manque de rapidité serait de conserver l'élan de vos pas précédents, qui peut être comparé au fait de commencer à courir dans la direction de la descente la plus forte et de se laisser guider par votre inertie le long de la pente. Vos yeux sont toujours bandés, vous risquez de prendre une mauvaise direction.

Ces deux analogies décrivent plus ou moins les algorithmes d'optimisation les plus utilisés : la descente de gradient et la descente de gradient accélérée. Ces deux algorithmes fonctionnent avec une connaissance précise du paysage local de notre montagne (c'est-à-dire de la fonction à minimiser).

Il est facile d'arriver à la conclusion que si l'on pouvait enlever son bandeau, on serait beaucoup plus rapide. Il est évident que si c'était possible, quelqu'un aurait déjà trouvé un algorithme pour le faire [3].

C'est là que l'optimisation multiniveau peut entrer en jeu. Pour poursuivre l'analogie, cela équivaudrait à enlever le bandeau sur les yeux tout en restant myope (ce n'est pas tout à fait la situation idéale, mais cela reste un progrès).

Il n'est pas nécessaire de connaître chaque rocher, chaque brin d'herbe, pour déduire une direction de descente, une connaissance approximative de la pente de la montagne est parfois suffisante. On peut alors faire de plus grands pas et atteindre plus rapidement le fond de la vallée.

L'essence de l'optimisation multiniveau consiste à utiliser une connaissance approximative du paysage de la fonction à minimiser, afin d'accélérer la convergence de l'algorithme d'optimisation sous-jacent. Comme nous allons le voir dans ce manuscrit, pour des problèmes d'optimisation classiques, tant que la fonction à minimiser possède une certaine structure, il est possible d'obtenir cette connaissance approximative et de l'exploiter.

Malheureusement, rien n'est gratuit en optimisation, et la construction et l'utilisation de cette connaissance approximative, pour accéler l'optimisation, ont un coût. Il faut donc faire un compromis. Tous les problèmes ne doivent pas être traités avec un algorithme multiniveau ; et tous les problèmes qui peuvent être traités avec un algorithme multiniveau, ne peuvent l'être sans une construction minutieuse.

L'objectif de cette thèse est de fournir des lignes directrices sur la construction d'algorithmes

[2] J'aurais aimé affirmer que c'est une bonne analogie pour expliquer les algorithme multiniveaux, mais je laisse cette décision au lecteur.

[3] En fait, dans certains contextes, on peut prouver qu'un tel algorithme n'existe pas [7].

multiniveaux pour l'optimisation non lisse. Grâce à ces connaissances, nous proposons un nouvel algorithme multiniveau, IML FISTA (Inexact Multilevel Fast Iterative Soft Thresholding Algorithm), avec des garanties de convergence au niveau de celles de l'état de l'art, et nous montrons son efficacité sur une large gamme de problèmes d'imagerie, de la reconstruction d'images couleur à la reconstruction d'images hyperspectrales. L'étude théorique et pratique de l'algorithme IML FISTA a suscité plusieurs questions, et nous présentons à la fin de ce manuscrit une nouvelle perspective sur les algorithmes multiniveaux, du point de vue des algorithmes de descente en bloc-coordonnée, qui a permis de répondre à certaines de ces questions.

Parmi nos nombreuses expériences numériques, nous développons une version d'IML FISTA qui peut être appliquée à des problèmes d'imagerie à grande échelle en radio-interférométrie. Pour mieux illustrer le potentiel d'IML FISTA, nous présentons dans la suite un résumé de notre contribution à ce problème d'imagerie.

Une application de l'optimisation multiniveau : l'imagerie radio-interférométrique

Les efforts déployés pour comprendre la formation des galaxies, des étoiles, des exoplanètes et de l'univers ont conduit au développement de nouvelles méthodes d'imagerie, et de techniques de calcul plus intensives pour traiter le volume de données généré.

Défi du passage à l'échelle en astronomie. Chaque jour, les appareils d'observation astronomiques collectent une énorme quantité de données, qui doivent être traitées. Dans le domaine optique, le télescope spatial James Webb (JWST), récemment lancé, produit des dizaines, voire des centaines de gigaoctets de données par jour [8], contre 1 ou 2 pour Hubble[4]. Dans le domaine radio, le Square Kilometer Array (SKA), une fois livré, devrait produire cinq téraoctets de données par seconde [9, 10]. Ces deux domaines de l'observation fournissent des informations précieuses et complémentaires sur les objets astronomiques (voir la figure 7.2).

Cela nécessite le développement d'algorithmes d'optimisation capables de passer à l'échelle avec des garanties de convergence solides. Les algorithmes multiniveaux sont l'une des nombreuses solutions à ce défi.

Dans cette section, nous proposons d'illustrer l'efficacité de la méthode que nous avons proposée dans cette thèse, et où les intuitions de la section précédente peuvent nous mener dans la compréhension des choix effectués. Pour ce faire, nous présentons un problème d'imagerie abordé dans cette thèse, qui est la reconstruction d'images à partir de données obtenues par radio-interférométrie [11]. Les prochains paragraphes constituent un aperçu de ce que nous avons fait sur ce problème, et une discussion approfondie est reportée au chapitre 6.

Radio-astronomie. Complémentaire de l'astronomie optique, la radio-astronomie est le domaine de l'astronomie qui étudie les objets dans le domaine des radio-fréquences, en collectant des ondes radio par l'intermédiaire de plusieurs antennes. La radio-interférométrie

[4] https://spectrum.ieee.org/james-webb-telescope-communications

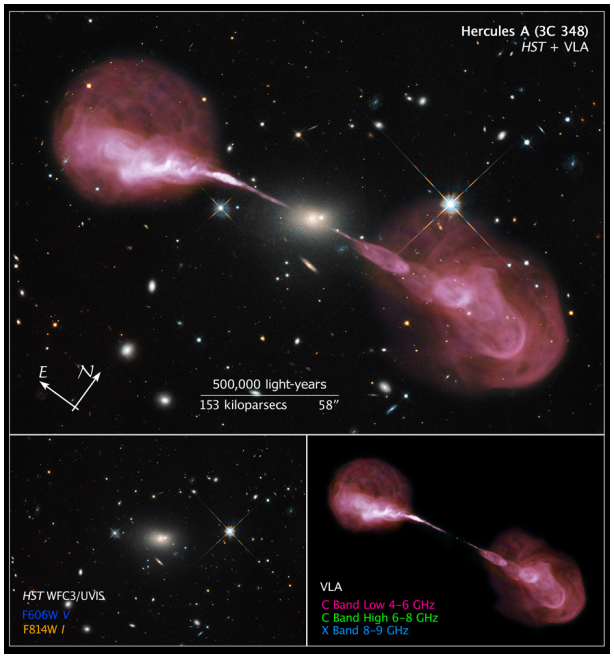

Figure 7.2: Comparaison entre une image optique (en bas à gauche) et une image radio (en bas à droite) de la même région du ciel : la galaxie Hercules A (3C48). Les deux images sont combinées en haut. L'image dans le visible a été obtenue par le télescope Hubble, tandis que l'image dans le domaine radio a été obtenue par le Karl G. Jansky Very Large Array (VLA). Crédits : NASA, ESA, S. Baum et C. O'Dea (RIT), R. Perley et W. Cotton (NRAO/AUI/NSF), et Hubble Heritage Team (STScI/AURA).

est une technique utilisée en radio-astronomie pour combiner les informations collectées par ces antennes afin d'obtenir des images du ciel avec une sensibilité, et une résolution, élevées, ce qui serait impossible avec une seule antenne.

Depuis les années 1950, les astronomes ont réussi à exploiter les techniques d'interférométrie pour surmonter les limites de la diffraction. L'interférométrie avait déjà une histoire bien développée à ce moment-là (voir [11]), ce qui a conduit au développement des radio-interféromètres : des réseaux constitués de plusieurs antennes de petit diamètre D, réparties sur une grande surface, et se comportant comme une seule antenne dont le diamètre apparent $\tilde{D}$ serait la plus grande distance entre deux antennes. Théoriquement, cela permet d'atteindre la résolution associée à une grande antenne de diamètre $\tilde{D}$(voir Figure 7.3).

La distance entre deux antennes est appelée "baseline". Dans la pratique, les astronomes combinent plusieurs paires d'antennes pour obtenir plusieurs "baseline" et pouvoir ainsi sonder le ciel dans plusieurs configurations. Les mesures obtenues de cette manière par les radio-interféromètres sont appelées *visibilités complexes* et recouvrent de manière inégale l'espace de Fourier. Cette technique ne permet de sonder le ciel que de manière éparse, ce qui nécessite l'utilisation de techniques de reconstruction d'images pour atteindre cette résolution.

Une de nos contributions, dans cette thèse, a été le développement d'un algorithme multiniveau adapté à l'imagerie radio-interférométrique.

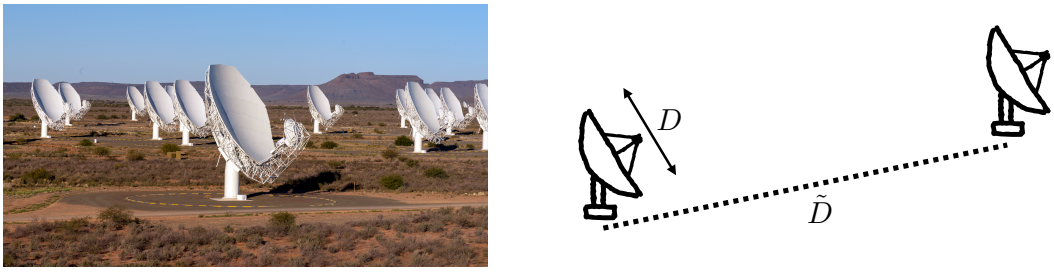

Figure 7.3: (À gauche) Le réseau radio-interférométrique MeerKAT en Afrique du Sud. Il se compose de 64 antennes et fera partie du futur réseau SKA. (À droite) Représentation schématique d'un réseau radio-interférométrique.

Optimisation multiniveau pour l'imagerie radio-interférométrique

Le nombre de visibilités complexes dans un problème d'imagerie radio-interférométrique est le principal goulot d'étranglement pour l'algorithme d'optimisation. Plus de données signifie plus de visibilités, et donc un coût de calcul plus élevé. L'algorithme multiniveau que nous proposons peut réduire ce coût en construisant une approximation grossière de la fonction à minimiser.

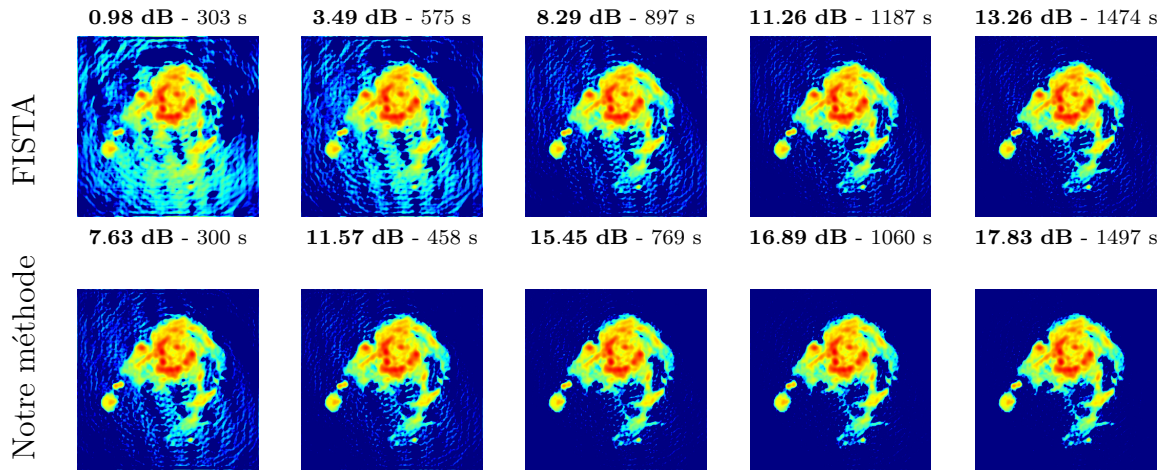

Figure 7.4: Reconstruction en échelle logarithmique d'une région de la galaxie M31 par FISTA (haut) et par notre méthode (bas) à des temps CPU équivalents. La légende en haut de chaque vignette se lit comme suit : **log SNR** en dB - temps CPU en secondes. Log SNR $= \mathrm{SNR}(\log_{10}(10^3 x + 1)/3, \log_{10}(10^3 x_{\mathrm{truth}} + 1)/3)$.

Pour se ramener à notre intuition, nous n'avons pas besoin de toutes les visibilités pour évaluer si notre reconstruction va dans la bonne direction. Une idée naturelle est donc de concevoir une connaissance approximative de la fonction objectif à minimiser en prenant en compte moins de visibilités. Nous sélectionnons un sous-ensemble de toutes les visibilités pour former un modèle grossier du problème. Dans cet exemple, nous prenons les visibilités les plus proches du centre du plan de Fourier, où se concentre la majeure partie de l'énergie du signal. Les composantes au centre du plan de Fourier sont des

composantes de basse fréquence et nous donnent une connaissance approximative de la fonction à minimiser. Les composantes à plus haute fréquence, qui sont les plus éloignées du centre, sont nos rochers et nos brins d'herbe. Ils peuvent donc être ignorés de temps à autre.

Avec un tèl modèle grossier, nous arrivons à accélérer la convergence de l'algorithme d'optimisation vers la solution du problème. En outre, nous obtenons de bonnes reconstructions de l'image en un temps de calcul bien plus court : *IML FISTA est* 3 *à* 5 *fois plus rapide que les algorithmes de l'état de l'art* (voir Figure 7.4 et le chapitre 6).

Résumé des chapitres

Résumé du Chapitre 2

Dans ce chapitre, nous présentons les outils nécessaires à la compréhension du manuscrit. Nous commençons par introduire les problèmes inverses dans le contexte des images, avant de présenter leurs formulations en tant que problèmes d'optimisation. Nous décrivons en en détail les deux termes qui composent la fonction objectif, l'attache aux données et la régularisation.

Ensuite, nous présentons les concepts classiques de l'optimisation convexe, afin d'introduire les algorithmes d'optimisation du premier ordre, descente de gradient et descente de gradient proximal, pour minimiser les fonctions qui nous intéressent dans cette thèse. Cette présentation passe par les notions de directions de descente pour les fonctions lisses et non-lisses, ainsi que du gradient et de l'opérateur proximal.

Au passage, nous présentons différentes notions de convergence vers une solution, qui nous permettrons de mieux caractériser l'efficacité des algorithmes. Nous terminons ce chapitre par une rapide présentation des stratégies permettant d'accélérer ces algorithmes du premier ordre.

Résumé du Chapitre 3

Dans ce chapitre, nous introduisons les méthodes multiniveaux. Ces méthodes étant considérées comme le standard en résolution d'équations aux dérivées partielles (EDP), sous le nom de multi-grilles, nous commençons par un petit exemple pour illustrer leur attrait pratique et théorique dans ce contexte. En quelques mots, en jouant sur la taille de la grille sur laquelle on résout l'EDP, on peut rapidement voir que l'on peut utiliser des grilles plus grossières pour diminuer le coût en temps de calcul de résolution. En plus, les méthodes itératives qui étaient communément utilisées avant l'introduction du multi-grilles, ont des taux de convergence théoriques qui dépendent des fréquences de l'erreur (l'écart avec la vraie solution de l'EDP): très rapide pour réduire les hautes fréquences, mais très lent pour réduire les basses fréquences.

Dès lors, il est intéressant de passer à des grilles plus grossières pour réduire les basses fréquences, et c'est ce que font les méthodes multi-grilles pour les EDP avec succès.

Il a donc été naturel de se demander si de telles performances pouvaient s'étendre à l'optimisation. De nombreux travaux ont été faits dans cette direction, que nous présentons. Nous terminons ce chapitre en présentant deux points bloquants identifiés par nous et la littérature, au succès des méthodes multiniveaux en optimisation.

Résumé du Chapitre 4

Dans ce chapitre nous présentons la contribution centrale de la thèse, notre algorithme Multi-niveaux FISTA Inexact. Notre objectif est de proposer un algorithme multiniveau avec les garanties de convergence de l'état de l'art (c'est-à-dire convergence des valeurs de la fonction vers une valeur minimale en $O(1/k^2)$, où k est le nombre d'itérations et convergence vers un minimiseur de la fonction), et capable de traiter les régularisations de l'état de l'art (qui n'ont pas d'opérateur proximal sous forme fermée) au niveau fin et au niveau grossier.

Nous présentons successivement les outils qui vont nous permettre: de passer de l'optimisation multiniveau lisse à l'optimisation multiniveau non-lisse, inspirés par [89,90]; de traiter les régularisations non-proximables; et enfin d'ajouter les pas d'extrapolation. Ensuite, IML FISTA est présenté dans le cadre le plus général possible, en décrivant les niveaux grossiers possibles (lisses et non-lisses), ainsi que la preuve des garanties de convergence précédemment citées. Nous terminons ce chapitre par une comparaison poussée de notre algorithme avec ceux de la littérature capables de traiter l'optimisation non-lisse. En quelques mots, IML FISTA est l'algorithme multiniveau le plus général possible, avec les meilleures garanties de convergence.

Résumé du Chapitre 5

Dans ce chapitre, nous présentons les applications de notre algorithme IML FISTA à une série de problèmes classiques en imagerie couleur et hyperspectrale. L'enjeu majeur d'un algorithme multiniveau étant la construction de niveaux grossiers efficaces, nous la discutons en détail et proposons une construction adaptée à l'imagerie.

Ensuite, nous basculons sur la présentation des résultats expérimentaux. Dans la continuité de la construction théorique des niveaux grossiers, nous avons testé un grand nombre de configurations de l'algorithme afin d'en identifier une qui soit robuste et efficace. Cette configuration a ensuite été utilisée pour traiter les problèmes suivants:

- défloutage et débruitage d'image en couleur (avec une réguarisation TV);
- reconstruction de pixels manquants dans une image en couleur (avec une régularisation NLTV);
- défloutage, débruitage et reconstruction de pixels manquants dans une image hyperspectrale (avec une régularisation NLTV).

Les deux premiers jeux d'expériences numériques nous ont permis de mettre en lumière le potentiel d'accélération d'IML FISTA par rapport aux algorithmes de l'état de l'art, ainsi que d'illustrer l'impact de la dimension sur les performances du multiniveau.

La restauration d'image hyperspectrale a donné lieu à la comparaison de deux manières de réduire la dimension du problème pour construire les modèles grossiers, en réduisant la taille de chaque bande, ou en fusionnant les bandes entre elles en fonction de leurs longueurs d'onde respectives. La seconde approche est la plus efficace, bien que la réduction de la dimension soit moindre.

Résumé du Chapitre 6

Dans ce chapitre, nous présentons une application de notre algorithme à un problème d'imagerie radio-interférométrique. L'enjeu de ce problème est de reconstruire une image à partir d'un échantillonage de la transformée de Fourier spatiale d'un objet astronomique. Ce problème nous permet de confronter IML FISTA à un problème avec de nombreuses contraintes pour lequel plusieurs méthodes de résolution ont été proposées.

Nous commençons par présenter le problème d'imagerie radio-interférométrique, ce qui nous permet de mettre en lumière que dans ce cas-ci ce n'est plus la taille de l'image qui empêche le passage à l'échelle des algorithmes, mais bien la taille des observations. Dès lors, nous proposons de construire notre niveau grossier en sélectionnant certaines de ces observations, sans réduire la taille de l'image. Cette construction nous permet d'accélerer grandement la résolution du problème avec IML FISTA, et d'ouvrir la porte à d'autres applications de notre algorithme à ce type de problèmes, par exemple sur des données réelles.

Résumé du Chapitre 7

Dans ce dernier chapitre, nous revisitons la construction des algorithmes multiniveaux à travers le point de vue des algorithmes de descente par coordonnées. Pour expliciter au mieux nos motivations, nous débutons par un exemple simple qui nous permet d'illustrer l'équivalence entre approches multiniveaux et approches par blocs. Cette équivalence nous permet d'analyser l'approche multiniveau via les outils de la littérature sur la descente par coordonnées.

Ensuite, étant donné que l'algorithme de descente par coordonnées est induit par une mise à jour des blocs imitant celle effectuée par le multiniveau (hiérarchique, potentiellement en parallèle et déterminée en amont de la procédure d'optimisation), il nous faut développer une nouvelle analyse pour prouver la convergence de cet algorithme. Cette analyse repose sur la propriété de Kurdyka-Łojasiewicz, qui nous permet de prouver la convergence de l'algorithme de descente par blocs hiérarchique, et donc de l'algorithme multiniveau.

Nous terminons par généraliser l'exemple introductif au cas à L-niveaux, ce qui nous permet de mettre en lumière certains enseignements sur la construction des algorithmes multiniveaux. Une série d'expériences numériques vient illustrer ces enseignements, ainsi que la pertinence de l'approche par blocs développée dans ce chapitre, en regard de celles de l'état de l'art.

Conclusion

Dans cette thèse, nous présentons une étude détaillée des méthodes multiniveaux pour l'optimisation non lisse, avec une application aux problèmes de reconstruction d'images. En exploitant une hiérarchie d'approximations de la fonction objectif, les algorithmes multiniveaux peuvent accélérer la convergence des algorithmes d'optimisation.

Dans le chapitre 4, nous avons présenté un cadre général pour concevoir un algorithme multiniveau avec des garanties optimales de convergence, pour les problèmes d'optimisation qui peuvent être non lisses et non proximables. En particulier, nous avons

obtenu un taux de convergence de $O(1/k^2)$ pour les valeurs de la fonction objectif et la convergence vers un minimiseur. Ce cadre a conduit à IML FISTA, une variante multiniveau inexacte de FISTA.

Avec cet algorithme, nous avons pu montrer que nous pouvions accélérer la résolution de problèmes de reconstruction d'images, avec une vitesse jusqu'à 10 fois supérieure à celle des algorithmes de l'état de l'art. Dans le chapitre 5, nous avons appliqué notre algorithme aux problèmes suivants :

(i) défloutage d'images noir et blanc, régularisé par la transformée en ondelettes,

(ii) défloutage d'images couleur, régularisé avec la variation totale,

(iii) reconstruction de pixels manquants d'images couleur, régularisé avec la variation totale non-locale,

(iv) reconstruction de pixels manquants et défloutage d'images hyperspectrales, régularisé avec la variation totale non-locale,

et avons montré qu'il peut fournir une bonne accélération sur tous ces problèmes. Ces expériences nous ont permis de conclure que les méthodes multiniveaux sont des approches intéressantes pour accélérer la résolution des problèmes de reconstruction d'images et qu'elles peuvent être appliquées à un large éventail de problèmes.

C'est pourquoi nous avons décidé de nous attaquer à un problème d'imagerie plus réaliste dans le chapitre 6 : la reconstruction d'images radio-interférométriques (RI) régularisée avec la transformée en ondelettes et une contrainte de positivité. Nous avons montré que l'IML FISTA peut être appliqué avec succès à ce problème, offrant une accélération significative par rapport aux algorithmes de l'état de l'art. En outre, nous avons mis en œuvre une nouvelle façon de réduire la dimension du problème, en construisant notre hiérarchie de niveaux sur une sélection de moins en moins d'observations (visibilités) à chaque niveau, pour être plus en phase avec les défis de passage à l'échelle des problèmes de radio-interférométrie [169].

Nous avons également remarqué que les garanties théoriques d'IML FISTA ne sont toujours pas à la hauteur des performances que nous pouvons observer en pratique, ce qui est un bon signe pour la robustesse de la méthode, mais aussi une incitation à une analyse théorique plus approfondie de l'algorithme. Jusqu'à présent, notre expérience avec IML FISTA est cohérente avec l'expérience d'autres auteurs sur les méthodes multiniveaux, c'est-à-dire que la principale difficulté pour les faire fonctionner réside dans les détails de l'implémentation : choix des modèles grossiers, choix des algorithmes à chaque niveau, etc.

Pour illustrer ce phénomène avec un exemple, dans le contexte général d'IML FISTA, nous ne pouvons que garantir qu'une étape multiniveau diminue la fonction objectif au niveau fin avec une petite erreur. Dans nos expériences, nous n'avons jamais observé cette erreur, ce qui amène à se poser la question suivante : est-il possible de prouver qu'une correction multiniveau est toujours une direction de descente ?

Des progrès dans cette direction (et dans d'autres) ont été obtenus au chapitre 7, où nous avons étudié la convergence des algorithmes multiniveaux du point de vue des algorithmes de descente par blocs. Pour cette étude, nous avons fourni une preuve de convergence pour un nouvel algorithme hiérarchique de descente par bloc, applicable à

l'optimisation non lisse et non convexe. En ce qui concerne notre question, dans ce contexte, nous sommes en mesure de garantir qu'une étape multiniveau fait décroître la fonction objectif, même pour des problèmes non convexes.

Ce nouveau point de vue nous a conduits à une conception rigoureuse d'un algorithme à plusieurs niveaux pour le défloutage d'images, régularisé avec des ondelettes, qui a apporté une réponse aux questionnements théoriques que nous avions rencontrés avec IML FISTA dans ce contexte.

Le travail présenté dans ce manuscrit ouvre la voie à de nouveaux développements théoriques et pratiques.

Perspectives pratiques

Tout d'abord, du point de vue des applications, nos expériences numériques ont mis en évidence que les méthodes multiniveaux ont un grand potentiel pour les problèmes d'imagerie où une suite de problèmes d'optimisation doivent être résolus. Pour de telles applications, le gain des approches multiniveaux pourrait se cumuler à chaque résolution, conduisant à des accélérations encore plus importantes. Par exemple, en imagerie radio-interférométrique, le volume de données à traiter est si important qu'il est courant de reconstruire l'image via la résolution d'une suite de problèmes d'optimisation avec des ensembles d'observations disjoints, une reconstruction "en ligne" [186]. Le multiniveau pourrait être utilisé pour accélérer la résolution de chacun de ces problèmes d'optimisation. Pour poursuivre sur l'imagerie radio-interférométrique, évaluer les performances d'IML FISTA sur des données réelles serait une prochaine étape naturelle. Nous pourrions ensuite envisager d'étendre le cadre d'IML FISTA à la construction d'un algorithme multiniveau pour résoudre la version contrainte de SARA (voir Annexes A.3.2 et A.2.3).

Du point de vue de la restauration d'images, de nombreuses directions peuvent être explorées. Pour commencer avec les approches variationnelles, nous avons brièvement parlé de la variation totale généralisée (TGV) au Chapitre 2. TGV offre une régularisation plus souple que TV et est plus simple que NLTV tout en fournissant des résultats de reconstruction similaires. Le cadre de la TGV n'est pas immédiatement pris en compte dans celui de nos algorithmes multiniveaux, car elle n'est pas formulée comme la composition d'une norme et d'un opérateur linéaire [34]. Cela pourrait donc constituer une extension intéressante.

Avec l'introduction de techniques d'apprentissage profond pour la restauration d'images, il serait intéressant de voir comment ces techniques pourraient être combinées avec des algorithmes multiniveaux. Les récents et nombreux développements visant à entraîner les réseaux neuronaux profonds à imiter les régularisateurs variationnels tels que RED [39,40] ou les méthodes Plug-and-Play (PnP) [38], semblent être un bon point de départ. RED par exemple a une formulation lisse [39], qui consiste à écrire la régularisation sous forme suivante:

$$R(x) = g_\sigma(x), \tag{7.106}$$

dont le gradient est explicitement formulé comme suit : $\nabla g_\sigma : x \mapsto x - \mathrm{NN}_\sigma(x)$, où NN_σ est un réseau de neurones entraîné pour une tâche de débruitage, paramétré par un niveau de bruit σ. Les méthodes PnP, quant à elles, relient le réseau de neurones à l'opérateur proximal de la régularisation. Ainsi $\mathrm{prox}_{g_\sigma} = \mathrm{NN}_\sigma$ [41–43]. Dans les deux cas,

la méthode résultante est itérative, avec un coût élevé en temps de calcul par rapport aux approches variationnelles, car l'évaluation d'un réseau de neurones est très coûteuse [220][5]. Un algorithme multiniveau pourrait donc contribuer à réduire ce coût. Cependant, nous avons constaté dans nos propres expériences que les modèles grossiers doivent être efficaces pour fournir une accélération, nous devons donc construire une approximation astucieuse du réseau de neurones ou, par exemple, ne pas régulariser le niveau grossier comme dans le Chapitre 6 (tout en maintenant la cohérence du premier ordre). Des travaux sont actuellement menés dans cette direction.

Les résultats du dernier chapitre suggèrent également de revoir certaines de nos expériences numériques, en particulier la restauration d'images hyperspectrales. La notion de bande est analogue à la notion de blocs dans notre cadre théorique de la descente par coordonnées. Il pourrait être intéressant de définir un algorithme multiniveau qui exploiterait la hiérarchie intrinsèque des bandes [223, 224], et donc améliorerait peut-être la performance de l'algorithme.

L'application de principes similaires à de nouveaux problèmes d'optimisation pour construire des algorithmes multiniveaux de manière *ad hoc* pourrait permettre d'éviter une grande partie du processus d'essais et d'erreurs qui est actuellement nécessaire pour concevoir un algorithme multiniveau afin de résoudre un nouveau problème.

Perspectives théoriques

Pour IML FISTA. D'un point de vue théorique, même si nous avons réussi à faire quelques progrès dans l'analyse de la convergence des algorithmes multiniveaux dans le chapitre 7, un écart subsiste avec la généralité du cadre d'IML FISTA.

Premièrement, les corrections grossières sont garanties de diminuer la fonction objectif dans le cadre BCD, ce qui n'est pas garanti dans le cadre d'IML FISTA (ou dans tout autre cadre multiniveau général pour l'optimisation non lisse). Nous avons tenté d'améliorer le cadre théorique du lissage en Annexe A.2.1, en examinant une condition de décroissance suffisante, mais les résultats obtenus ne sont pas exploitables en pratique. La question reste donc ouverte.

Deuxièmement, les étapes proximales inexactes telles que caractérisées par [62] dans le cas convexe, ne sont pas (encore) possibles dans le cadre défini par la propriété de Kurdyka-Łojasiewicz, même si une certaine notion d'inexactitude existe déjà [212, 214]. Essayer d'étendre nos résultats de convergence obtenus dans le chapitre 7 à ce cadre serait donc une prochaine étape naturelle.

Enfin, sur une note plus positive, la preuve de convergence que nous avons utilisée au chapitre 4 peut être réutilisée pour prouver la convergence d'autres algorithmes multiniveaux, comme nous le démontrons en Annexe A.3.2. Nous pouvons donc nous concentrer sur la recherche d'implémentations efficaces de ces algorithmes, sans nous préoccuper de prouver la convergence.

[5]Pour citer les auteurs : « L'inférence des réseaux neuronaux représenterait 90 % du coût de l'apprentissage automatique à grande échelle selon des rapports indépendants de NVIDIA [221] et d'Amazon Web Services [222]. »

Pour la descente par coordonnées. Maintenant, du point de vue de la descente par coordonnées, on peut se demander si les résultats du chapitre 7 pourraient être prouvés dans un cadre complètement stochastique (non convexe). Est-il possible de considérer des corrélations entre les mises à jour des blocs d'une itération à l'autre tout en conservant les mêmes garanties que dans le cadre Féjer de [196] ? Une réponse à cette question améliorerait grandement les garanties de convergence de certaines méthodes de descente par coordonnées stochastiques.

Il a été prouvé dans [179] que la procédure de repondération mise en œuvre dans SARA [167] pouvait être interprétée comme la résolution unique d'un problème d'optimisation à l'aide d'un algorithme de descente par blocs [181]. La nouvelle perspective sur les algorithmes multiniveaux, que nous apporte le cadre de la descente par coordonnées, pourrait être utilisée pour analyser la convergence d'un algorithme multiniveau appliqué à ce problème.

De plus, il est connu que les algorithmes BCD sont vraiment compétitifs dans les contextes où la mise à jour de toutes les coordonnées à la fois n'est pas possible [192]. Il serait donc intéressant de voir comment nous pourrions adapter notre sélection hiérarchique des blocs à mettre à jour dans de tels contextes.

Enfin, nous n'avons pas étudié l'impact de l'ajout d'inertie [190] sur la convergence de notre algorithme H-BC-FB. Une telle étude nous aiderait à établir l'analogie entre IML FISTA et H-BC-FB, et pourrait peut-être fournir des indications sur la manière d'améliorer la convergence d'IML FISTA : devrions-nous utiliser des pas d'extrapolations dans les étapes multiniveaux ?

Optimisation d'ordre supérieur. Les méthodes d'optimisation d'ordre supérieur sont connues pour mieux s'adapter à la géométrie de la fonction et donc converger plus rapidement, mais à un coût plus élevé [55, 225]. Les méthodes du premier ordre sont mieux comprises et plus utilisées que les méthodes d'ordre supérieur, et leur potentiel est très probablement proche d'avoir été pleinement exploité aujourd'hui. Par conséquent, des efforts devraient être faits pour réduire le coût de calcul des méthodes d'optimisation d'ordre supérieur.

Du point de vue de l'optimisation multiniveau non lisse, un premier pas dans cette direction pourrait être fait en essayant d'imiter la cohérence du second ordre avec le terme d'attache aux données, en construisant une approximation de Galerkin de la matrice hessienne au niveau grossier. Cela peut être implicitement fait en choisissant $\mathrm{A}_H = I_h^H \mathrm{A}_h I_H^h$ dans nos problèmes de restauration d'image, comme $\mathrm{A}_H^T \mathrm{A}_H = I_h^H \mathrm{A}_h^T I_H^h I_h^H \mathrm{A}_h I_H^h = I_h^H \mathrm{A}_h^T \mathrm{A}_h I_H^h$ si $I_H^h I_h^H = \mathrm{Id}_H$ (l'opérateur identité au niveau grossier). Ceci est trivialement satisfait dans le cadre BCD de l'algorithme multiniveau pour le défloutage régularisé par ondelettes. L'étude des valeurs propres de $I_h^H \mathrm{A}_h^T \mathrm{A}_h I_H^h$ et de leur relation avec celles de $\mathrm{A}_h^T \mathrm{A}_h$ pourrait nous indiquer dans quelle mesure les informations de second ordre du niveau grossier sont fidèles à celles du niveau fin, sans qu'il soit nécessaire d'envoyer la matrice hessienne au niveau grossier à chaque étape multiniveau.

Ces idées pourraient être incorporées dans un algorithme de Newton proximal à plusieurs niveaux [226] qui utilise la matrice hessienne du terme lisse dans les itérations, pour mieux adapter le pas de gradient à la courbure de la fonction.

Sommaire en français

Bibliography

[1] A. N. Tikhonov *et al.*, "On the stability of inverse problems," in *Dokl. akad. nauk sssr*, vol. 39, pp. 195–198, 1943.

[2] M. Bertero, P. Boccacci, and C. De Mol, *Introduction to inverse problems in imaging*. CRC press, 2021.

[3] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images*. SIAM, 2006.

[4] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[5] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, *et al.*, "An introduction to total variation for image analysis," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, no. 263-340, p. 227, 2010.

[6] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Foundations and Trends® in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[7] Y. Nesterov *et al.*, *Lectures on convex optimization*, vol. 137. Springer, 2018.

[8] A. Johns, B. Seaton, J. Gal-Edd, R. Jones, C. Fatig, and F. Wasiak, "James webb space telescope: L2 communications for science data processing," in *Observatory Operations: Strategies, Processes, and Systems II*, vol. 7016, pp. 425–431, SPIE, 2008.

[9] J. Birdi, *Advanced sparse optimization algorithms for interferometric imaging inverse problems in astronomy*. PhD thesis, Heriot-Watt University, 2019.

[10] P. C. Broekema, R. V. van Nieuwpoort, and H. E. Bal, "The square kilometre array science data processor. preliminary compute platform design," *Journal of Instrumentation*, vol. 10, no. 07, p. C07004, 2015.

[11] A. R. Thompson, J. M. Moran, and G. W. Swenson, *Interferometry and synthesis in radio astronomy*. Springer Nature, 2017.

[12] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numerica*, vol. 25, pp. 161–319, 2016.

[13] L. Rudin, S. J. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[14] R. Abergel, C. Louchet, L. Moisan, and T. Zeng, "Total variation restoration of images corrupted by poisson noise with iterated conditional expectations," in *Scale Space and Variational Methods in Computer Vision: 5th International Conference, SSVM 2015, Lège-Cap Ferret, France, May 31-June 4, 2015, Proceedings 5*, pp. 178–190, Springer, 2015.

[15] S. Setzer, G. Steidl, and T. Teuber, "Deblurring poissonian images by split bregman techniques," *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 193–199, 2010.

[16] H. Talbot, H. Phelippeau, M. Akil, and S. Bara, "Efficient poisson denoising for photography," pp. 3881 – 3884, 12 2009.

[17] C. Deledalle, F. Tupin, and L. Denis, "Poisson NL means: Unsupervised non local means for poisson noise," in *Proceedings of the International Conference on Image Processing*, pp. 801–804, 2010.

[18] D. Geman and S. Geman, "Bayesian image analysis," in *Disordered systems and biological organization*, pp. 301–319, Springer, 1986.

[19] J. Besag, J. York, and A. Mollié, "Bayesian image restoration, with two applications in spatial statistics," *Annals of the institute of statistical mathematics*, vol. 43, pp. 1–20, 1991.

[20] N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet, "Wavelet-based image deconvolution and reconstruction," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, Feb 2016. Tutorial paper.

[21] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[22] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[23] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[24] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[25] M. A. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[26] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[27] P. Combettes and V. Wajs, "Signal Recovery by Proximal Forward-Backward Splitting," *SIAM Multiscale Model. Simul.*, vol. 4, pp. 1168–1200, 2005.

[28] L. Jacques, L. Duval, C. Chaux, and G. Peyré, "A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity," *Signal Processing*, vol. 91, no. 12, pp. 2699–2730, 2011.

[29] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, pp. 167–188, 1997.

[30] G. Steidl and J. Weickert, "Relations between soft wavelet shrinkage and total variation denoising," in *Joint pattern recognition symposium*, pp. 198–205, Springer, 2002.

[31] U. Kamilov, E. Bostan, and M. Unser, "Generalized total variation denoising via augmented lagrangian cycle spinning with haar wavelets," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 909–912, Ieee, 2012.

[32] P. Rodríguez, "Total variation regularization algorithms for images corrupted with different noise models: a review," *Journal of Electrical and Computer Engineering*, vol. 2013, no. 1, p. 217021, 2013.

[33] G. Chierchia, N. Pustelnik, B. Pesquet-Popescu, and J.-C. Pesquet, "A Non-Local Structure Tensor Based Approach for Multicomponent Image Recovery Problems," *IEEE Trans. Image Process.*, vol. 23, pp. 5531–5544, Dec. 2014. arXiv:1403.5403.

[34] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[35] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. Bach, "Supervised dictionary learning," *Advances in neural information processing systems*, vol. 21, 2008.

[36] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE transactions on medical imaging*, vol. 30, no. 5, pp. 1028–1041, 2010.

[37] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[38] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE global conference on signal and information processing*, pp. 945–948, IEEE, 2013.

[39] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[40] R. Cohen, M. Elad, and P. Milanfar, "Regularization by denoising via fixed-point projection (red-pro)," *SIAM Journal on Imaging Sciences*, vol. 14, no. 3, pp. 1374–1406, 2021.

[41] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *International Conference on Machine Learning*, pp. 5546–5557, PMLR, 2019.

[42] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux, "Learning maximally monotone operators for image recovery," *SIAM Journal on Imaging Sciences*, vol. 14, no. 3, pp. 1206–1237, 2021.

[43] S. Hurault, A. Leclaire, and N. Papadakis, "Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization," in *International Conference on Machine Learning*, pp. 9483–9505, PMLR, 2022.

[44] M. Terris, A. Dabbech, C. Tang, and Y. Wiaux, "Image reconstruction algorithms in radio interferometry: From handcrafted to learned regularization denoisers," *Monthly Notices of the Royal Astronomical Society*, vol. 518, no. 1, pp. 604–622, 2023.

[45] Z. Wang and E. P. Simoncelli, "Reduced-reference image quality assessment using a wavelet-domain natural image statistic model," in *Human Vision and Electronic Imaging X, Proc. SPIE*, vol. 5666, (San Jose, CA), p. 18 March 2005, 2005.

[46] P. Terhorst, J. N. Kolf, N. Damer, F. Kirchbuchner, and A. Kuijper, "SER-FIQ: Unsupervised estimation of face image quality based on stochastic embedding robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5651–5660, 2020.

[47] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Rankiqa: Learning from rankings for no-reference image quality assessment," in *Proceedings of the IEEE international conference on computer vision*, pp. 1040–1049, 2017.

[48] 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*, vol. 13, no. 4, pp. 600–612, 2004.

[49] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 20th international conference on pattern recognition*, pp. 2366–2369, IEEE, 2010.

[50] R. T. Rockafellar, *Convex Analysis*. 1970.

[51] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms I: Fundamentals*, vol. 305. Springer science & business media, 1996.

[52] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317. Springer Science & Business Media, 2009.

[53] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics, New York: Springer International Publishing, 2017.

[54] C. Zalinescu, *Convex analysis in general vector spaces*. World scientific, 2002.

[55] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[56] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.

[57] A. Beck and M. Teboulle, "Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems," *IEEE Trans. Image Process.*, vol. 18, pp. 2419–2434, Nov. 2009.

[58] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 2nd ed., 1999.

[59] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, vol. 30 of *Classics in Applied Mathematics*. Philadelphia: SIAM, 2000.

[60] L. M. Briceño-Arias and N. Pustelnik, "Theoretical and numerical comparison of first order algorithms for cocoercive equations and smooth convex optimization," *Signal Processing*, vol. 206, p. 108900, 2023.

[61] G. Garrigos, *Descent dynamical systems and algorithms for tame optimization and multi-objective problems.* PhD thesis, Université de Montpellier; Universidad Tecnica Federico Santa Maria, 2015.

[62] S. Villa, S. Salzo, L. Baldassarre, and A. Verri, "Accelerated and Inexact Forward-Backward Algorithms," *SIAM Journal on Optimization*, vol. 23, pp. 1607–1633, Jan. 2013.

[63] P. L. Combettes, Đ. Dũng, and B. C. Vũ, "Dualization of signal recovery problems," *Set-Valued and Variational Analysis*, vol. 18, no. 3-4, pp. 373–404, 2010.

[64] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[65] R. M. Gower, "Convergence theorems for gradient descent," *Lecture notes for Statistical Optimization*, 2018.

[66] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 2004.

[67] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," *Soviet Math. Dokl.*, vol. 27, pp. 372–376, 1983.

[68] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, pp. 183–202, Jan. 2009.

[69] A. Chambolle and C. H. Dossal, "On the convergence of the iterates of "FISTA"," *Journal of Opt. Theory and Applications*, vol. 166, no. 3, p. 25, 2015.

[70] J.-F. Aujol and C. Dossal, "Stability of Over-Relaxations for the Forward-Backward Algorithm, Application to FISTA," *SIAM Journal on Optimization*, vol. 25, pp. 2408–2433, Jan. 2015.

[71] J.-F. Aujol, C. Dossal, and A. Rondepierre, "Fista is an automatic geometrically optimized algorithm for strongly convex functions," *Mathematical Programming*, vol. 204, no. 1, pp. 449–491, 2024.

[72] J.-F. Aujol, C. Dossal, and A. Rondepierre, "Optimal convergence rates for Nesterov acceleration," *SIAM Journal on Optimization*, vol. 29, no. 4, pp. 3131–3153, 2019.

[73] J.-F. Aujol, C. Dossal, H. Labarrière, and A. Rondepierre, "FISTA restart using an automatic estimation of the growth parameter," 2022.

[74] S. Rebegoldi and L. Calatroni, "Scaled, inexact, and adaptive generalized fista for strongly convex optimization," *SIAM Journal on Optimization*, vol. 32, no. 3, pp. 2428–2459, 2022.

[75] V. Apidopoulos, J.-F. Aujol, and C. Dossal, "Convergence rate of inertial Forward-Backward algorithm beyond Nesterov's rule," *Mathematical Programming*, vol. 180, pp. 137–156, 2020.

[76] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.

[77] J.-F. Aujol and C. Dossal, "Optimal rate of convergence of an ODE associated to the fast gradient descent schemes for b> 0," 2017.

[78] M. Muehlebach and M. Jordan, "A dynamical systems perspective on nesterov acceleration," in *International Conference on Machine Learning*, pp. 4656–4662, PMLR, 2019.

[79] 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*, vol. 162, no. 1, pp. 107–132, 2014.

[80] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward–backward algorithm," *Journal of Global Optimization*, vol. 66, no. 3, pp. 457–485, 2016.

[81] S. Salzo, "The variable metric forward-backward splitting algorithm under mild differentiability assumptions," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2153–2181, 2017.

[82] J. F. Bard, *Practical bilevel optimization: algorithms and applications*, vol. 30. Springer Science & Business Media, 2013.

[83] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of operations research*, vol. 153, pp. 235–256, 2007.

[84] W. Hackbusch, *Multi-grid methods and applications*. Springer, 1985.

[85] S. F. McCormick, *Multigrid methods*. SIAM, 1987.

[86] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial, Second Edition*. Society for Industrial and Applied Mathematics, second ed., Jan. 2000.

[87] T. Gerya, *Introduction to Numerical Geodynamic Modelling*. Cambridge: Cambridge University Press, 2010.

[88] A. Ang, H. De Sterck, and S. Vavasis, "MGProx: A nonsmooth multigrid proximal gradient method with adaptive restriction for strongly convex optimization," *SIAM Journal on Optimization*, vol. 34, no. 3, pp. 2788–2820, 2024.

[89] V. Hovhannisyan, P. Parpas, and S. Zafeiriou, "MAGMA: Multilevel Accelerated Gradient Mirror Descent Algorithm for Large-Scale Convex Composite Minimization," *SIAM Journal on Imaging Sciences*, vol. 9, pp. 1829–1857, Jan. 2016.

[90] P. Parpas, "A Multilevel Proximal Gradient Algorithm for a Class of Composite Optimization Problems," *SIAM Journal on Scientific Computing*, vol. 39, no. 5, pp. S681–S701, 2017.

[91] R. S. Varga, "Iterative analysis," *New Jersey*, vol. 322, 1962.

[92] R. P. Fedorenko, "A relaxation method for solving elliptic difference equations," *USSR Computational Mathematics and Mathematical Physics*, vol. 1, pp. 1092–1096, 1962.

[93] R. V. Southwell, "Relaxation methods in engineering science," *The Journal of the Royal Aeronautical Society*, vol. 45, no. 364, pp. 176–178, 1941.

[94] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*. Elsevier, 2000.

[95] W. Hackbusch, "Convergence of multigrid iterations applied to difference equations," *Mathematics of Computation*, vol. 34, pp. 425–440, 1980.

[96] S. G. Nash, "A Multigrid Approach to Discretized Optimization Problems," *Optimization Methods and Software*, vol. 14, no. 1-2, pp. 99–116, 2000.

[97] P. Deuflhard and M. Weiser, "Local inexact newton multilevel fem for nonlinear elliptic problems," in *Computational Science for the 21st Century*, 1997.

[98] S. Gratton, A. Sartenaer, and P. L. Toint, "Recursive trust-region methods for multiscale nonlinear optimization," *SIAM Journal on Optimization*, vol. 19, no. 1, pp. 414–444, 2008.

[99] M. Donatelli, "An algebraic generalization of local Fourier analysis for grid transfer operators in multigrid based on Toeplitz matrices," *Numerical Linear Algebra with Applications*, vol. 17, pp. 179–197, Apr. 2010.

[100] M. Donatelli, "An Iterative Multigrid Regularization Method for Toeplitz Discrete Ill-Posed Problems," *Numerical Mathematics: Theory, Methods and Applications*, vol. 5, pp. 43–61, June 2012.

[101] A. Braides, "A handbook of Gamma-convergence," in *Handbook of Differential Equations: stationary partial differential equations*, vol. 3, pp. 101–213, Elsevier, 2006.

[102] H. Calandra, S. Gratton, E. Riccietti, and X. Vasseur, "On High-Order Multilevel Optimization Strategies," *SIAM Journal on Optimization*, vol. 31, no. 1, pp. 307–330, 2021.

[103] Z. Wen and D. Goldfarb, "A Line Search Multigrid Method for Large-Scale Nonlinear Optimization," *SIAM Journal on Optimization*, vol. 20, pp. 1478–1503, Jan. 2010.

[104] R. M. Lewis and S. G. Nash, "Model problems for the multigrid optimization of systems governed by differential equations," *SIAM Journal on Scientific Computing*, vol. 26, no. 6, pp. 1811–1837, 2005.

[105] R. M. Lewis and S. G. Nash, "Using inexact gradients in a multilevel optimization algorithm," *Computational Optimization and Applications*, vol. 56, no. 1, pp. 39–61, 2013.

[106] S. G. Nash, "Convergence and descent properties for a class of multilevel optimization algorithms," 2010.

[107] A. Javaherian and S. Holman, "A Multi-Grid Iterative Method for Photoacoustic Tomography," *IEEE Transactions on Medical Imaging*, pp. 696–706, Mar. 2017.

[108] J. Plier, F. Savarino, M. Kočvara, and S. Petra, "First-Order Geometric Multilevel Optimization for Discrete Tomography," in *Scale Space and Variational Methods in Computer Vision*, vol. 12679, pp. 191–203, Cham: Springer International Publishing, 2021. Series Title: Lecture Notes in Computer Science.

[109] S. W. Fung and Z. Wendy, "Multigrid Optimization for Large-Scale Ptychographic Phase Retrieval," *SIAM Journal on Imaging Sciences*, vol. 13, pp. 214–233, Jan. 2020.

[110] V. Hovhannisyan, Y. Panagakis, P. Parpas, and S. Zafeiriou, "Fast multilevel algorithms for compressive principal component pursuit," *SIAM Journal on Imaging Sciences*, vol. 12, no. 1, pp. 624–649, 2019.

[111] J. S. Campos and P. Parpas, "A multigrid approach to SDP relaxations of sparse polynomial optimization problems," *SIAM Journal on Optimization*, vol. 28, pp. 1–29, 2018.

[112] C. P. Ho, M. Kočvara, and P. Parpas, "Newton-type multilevel optimization method," *Optimization Methods and Software*, pp. 1–34, Dec. 2019.

[113] M. I. Espanol, *Multilevel methods for discrete ill-posed problems: Application to deblurring*. PhD thesis, Tufts University, 2009.

[114] M. I. Español and M. E. Kilmer, "Multilevel Approach For Signal Restoration Problems With Toeplitz Matrices," *SIAM Journal on Scientific Computing*, vol. 32, pp. 299–319, Jan. 2010.

[115] M. K. Ng, R. H. Chan, and W.-C. Tang, "A fast algorithm for deblurring models with neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 851–866, 1999.

[116] A. Buccini and M. Donatelli, "A multigrid frame based method for image deblurring," *Electronic Transactions on Numerical Analysis*, vol. 53, pp. 283–312, 2020.

[117] S. MacLachlan and Y. Saad, "A greedy strategy for coarse-grid selection," *SIAM Journal on Scientific Computing*, vol. 29, no. 5, pp. 1825–1853, 2007.

[118] I. Luz, M. Galun, H. Maron, R. Basri, and I. Yavneh, "Learning algebraic multigrid using graph neural networks," in *International Conference on Machine Learning*, pp. 6489–6499, PMLR, 2020.

[119] A. Taghibakhshi, S. MacLachlan, L. Olson, and M. West, "Optimization-based algebraic multigrid coarsening using reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 12129–12140, 2021.

[120] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE signal processing magazine*, vol. 14, no. 2, pp. 24–41, 1997.

[121] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," *arXiv preprint arXiv:1901.06523*, 2019.

[122] N. Rahaman, D. Arpit, A. Baratin, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of deep neural networks," in *International Conference on Machine Learning*, 2019.

[123] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[124] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[125] A. M. Teney, D.and Nicolicioiu, V. Hartmann, and E. Abbasnejad, "Neural redshift: Random networks are not random functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4786–4796, 2024.

[126] G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves, "IML FISTA: A Multilevel Framework for Inexact and Inertial Forward-Backward. Application to Image Restoration," *SIAM Journal on Imaging Sciences*, vol. 17, no. 3, pp. 1347–1376, 2024.

[127] G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves, "Multilevel Fista For Image Restoration," *IEEE ICASSP, Rhodes, Greece*, 4-10 June 2023.

[128] G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves, "Méthodes proximales multiniveaux pour la restauration d'images," *28ème colloque GRETSI*, Sept. 2022.

[129] A. Kopanicáková and R. Krause, "Globally convergent multilevel training of deep residual networks," *SIAM Journal on Scientific Computing*, vol. 45, no. 3, pp. S254–S280, 2023.

[130] B. Woodworth, K. Mishchenko, and F. Bach, "Two losses are better than one: Faster optimization using a cheaper proxy," in *International Conference on Machine Learning*, pp. 37273–37292, PMLR, 2023.

[131] D. L. Donoho, "For most large underdetermined systems of equations, the minimal ℓ_1-norm near-solution approximates the sparsest near-solution," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 7, pp. 907–934, 2006.

[132] A. Beck and M. Teboulle, "Smoothing and First Order Methods: A Unified Framework," *SIAM Journal on Optimization*, vol. 22, pp. 557–580, Jan. 2012.

[133] J.-J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *C. R. Acad. Sci.*, vol. 255, pp. 2897–2899, 1962.

[134] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[135] M. Schmidt, N. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[136] M. Barré, A. B. Taylor, and F. Bach, "Principled analyses and design of first-order methods with inexact proximal operators," *Mathematical Programming*, vol. 201, no. 1, pp. 185–230, 2023.

[137] P. L. Combettes and J.-C. Pesquet, "Proximal thresholding algorithm for minimization over orthonormal bases," *SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1351–1376, 2008.

[138] J.-C. Pesquet and N. Pustelnik, "A parallel inertial proximal optimization method," *Pacific Journal of Optimization*, vol. 8, no. 2, pp. 273–305, 2012.

[139] H. T. V. Le, N. Pustelnik, and M. Foare, "The faster proximal algorithm, the better unfolded deep learning architecture? the study case of image denoising," in *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 947–951, IEEE, 2022.

[140] L. Condat, "A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms," *Journal of Optimization Theory and Applications*, vol. 158, pp. 460–479, Aug. 2013.

[141] P. Machart, S. Anthoine, and L. Baldassarre, "Optimal Computational Trade-Off of Inexact Proximal Methods," Oct. 2012. arXiv:1210.5034.

[142] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[143] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Advances in Computational Mathematics*, vol. 38, no. 3, pp. 667–681, 2013.

[144] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, pp. 120–145, May 2011.

[145] J. Rasch and A. Chambolle, "Inexact first-order primal–dual algorithms," *Computational Optimization and Applications*, vol. 76, no. 2, pp. 381–430, 2020.

[146] J. Chen and I. Loris, "On starting and stopping criteria for nested primal-dual iterations," *Numerical algorithms*, vol. 82, pp. 605–621, 2019.

[147] S. Bonettini, M. Prato, and S. Rebegoldi, "A nested primal–dual fista-like scheme for composite convex optimization problems," *Computational Optimization and Applications*, vol. 84, no. 1, pp. 85–123, 2023.

[148] Z. Allen-Zhu and L. Orecchia, "Linear coupling: An ultimate unification of gradient and mirror descent," *arXiv preprint arXiv:1407.1537*, 2014.

[149] H.-C. Lai and L.-J. Lin, "Moreau-rockafellar type theorem for convex set functions," *Journal of Mathematical Analysis and Applications*, vol. 132, pp. 558–571, 1988.

[150] G. Lauga, E. Riccietti, N. Pustelnik, and P. Gonçalves, "Méthodes multi-niveaux pour la restauration d'images hyperspectrales," *29ème colloque GRETSI*, Sept. 2023.

[151] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, "Image inpainting: A review," *Neural Processing Letters*, vol. 51, pp. 2007–2028, 2020.

[152] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 3, pp. 730–742, 2018.

[153] L. Cheng, H. Wang, and Z. Zhang, "The solution of ill-conditioned symmetric toeplitz systems via two-grid and wavelet methods," *Computers & Mathematics with Applications*, vol. 46, pp. 793–804, Sept. 2003.

[154] L.-J. Deng, T.-Z. Huang, and X.-L. Zhao, "Wavelet-based two-level methods for image restoration," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 5079–5087, Dec. 2012.

[155] T. D. Luu, J. Fadili, and C. Chesneau, "Sampling from non-smooth distribution through Langevin diffusion," *preprint*, p. 27, 2017.

[156] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sensing*, vol. 12, no. 16, p. 2659, 2020.

[157] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.

[158] R. Pillay, J. Y. Hardeberg, and S. George, "Hyperspectral imaging of art: Acquisition and calibration workflows," *Journal of The American Institute for Conservation*, 2019.

[159] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot, "Noise reduction in hyperspectral imagery: Overview and application," *Remote Sensing*, vol. 10, no. 3, p. 482, 2018.

[160] J. Khodr and R. Younes, "Dimensionality reduction on hyperspectral images: A comparative review based on artificial datas," *International Congress on Image and Signal Processing*, Shangaï, China, October 15-17 2011.

[161] S. Jia, G. Tang, J. Zhu, and Q. Li, "A novel ranking-based clustering approach for hyperspectral band selection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 88–102, 2016.

[162] M. Golbabaee and P. Vandergheynst, "Hyperspectral image compressed sensing via low-rank and joint-sparse matrix recovery," in *2012 IEEE ICASSP*, pp. 2741–2744, Kyoto, Japon, 25 - 30 Mars 2012.

[163] P. A. Mitchell, "Hyperspectral digital imagery collection experiment (HYDICE)," in *Geographic Information Systems, Photogrammetry, and Geological/Geophysical Remote Sensing*, vol. 2587, pp. 70 – 95, International Society for Optics and Photonics, SPIE, 1995.

[164] M. A. Folkman, J. Pearlman, L. B. Liao, and P. J. Jarecke, "EO-1/hyperion hyperspectral imager design, development, characterization, and calibration," *Hyperspectral Remote Sensing of the Land and Atmosphere*, vol. 4151, pp. 40–51, 2001.

[165] D. H. Foster, K. Amano, S. M. C. Nascimento, and M. J. Foster, "Frequency of metamerism in natural scenes," *J. Opt. Soc. Am. A*, vol. 23, no. 10, pp. 2359–2372, 2006.

[166] G. Lauga, A. Repetti, E. Riccietti, N. Pustelnik, P. Gonçalves, and Y. Wiaux, "A multilevel framework for accelerating uSARA in radio-interferometric imaging,"

in *2024 32nd European Signal Processing Conference (EUSIPCO)*, pp. 2287–2291, 2024.

[167] R. E. Carrillo, J. D. McEwen, and Y. Wiaux, "Sparsity averaging reweighted analysis (SARA): a novel algorithm for radio-interferometric imaging," *Monthly Notices of the Royal Astronomical Society*, vol. 426, no. 2, pp. 1223–1234, 2012.

[168] J. A. Fessler and B. P. Sutton, "Nonuniform fast fourier transforms using min-max interpolation," *IEEE transactions on signal processing*, vol. 51, no. 2, pp. 560–574, 2003.

[169] A. Onose, R. E. Carrillo, A. Repetti, J. D. McEwen, J.-P. Thiran, J.-C. Pesquet, and Y. Wiaux, "Scalable splitting algorithms for big-data interferometric imaging in the SKA era," *Monthly Notices of the Royal Astronomical Society*, vol. 462, no. 4, pp. 4314–4335, 2016.

[170] Y. Mhiri, *Contributions aux méthodes de calibration et d'imagerie pour les radio-interféromètres en présence d'interférences*. PhD thesis, Université Paris-Saclay, 2023.

[171] A. Repetti, J. Birdi, A. Dabbech, and Y. Wiaux, "Non-convex optimization for self-calibration of direction-dependent effects in radio interferometric imaging," *Monthly Notices of the Royal Astronomical Society*, vol. 470, no. 4, pp. 3981–4006, 2017.

[172] A. Dabbech, A. Repetti, R. A. Perley, O. M. Smirnov, and Y. Wiaux, "Cygnus a jointly calibrated and imaged via non-convex optimization from vla data," *Monthly Notices of the Royal Astronomical Society*, vol. 506, no. 4, pp. 4855–4876, 2021.

[173] J. A. Högbom, "Aperture synthesis with a non-regular distribution of interferometer baselines," *Astronomy and Astrophysics Supplement, Vol. 15, p. 417*, vol. 15, p. 417, 1974.

[174] T. J. Cornwell, "Multiscale clean deconvolution of radio synthesis images," *IEEE Journal of selected topics in signal processing*, vol. 2, no. 5, pp. 793–801, 2008.

[175] S. Yatawatta, "Fundamental limitations of pixel based image deconvolution in radio astronomy," in *2010 IEEE Sensor Array and Multichannel Signal Processing Workshop*, pp. 69–72, IEEE, 2010.

[176] B. G. Clark, "An efficient implementation of the algorithm 'CLEAN'," *Astronomy and Astrophysics*, vol. 89, pp. 377–378, 1980.

[177] F. R. Schwab, "Relaxing the isoplanatism assumption in self-calibration; applications to low-frequency radio interferometry," *Astronomical Journal*, vol. 89, pp. 1076–1081, 1984.

[178] T. J. Cornwell, "A simple method of stabilizing the clean algorithm," *Astronomy and Astrophysics*, vol. 121, pp. 281–285, 1983.

[179] A. Repetti and Y. Wiaux, "Variable metric forward-backward algorithm for composite minimization problems," *SIAM Journal on Optimization*, vol. 31, no. 2, pp. 1215–1241, 2021.

[180] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier analysis and applications*, vol. 14, pp. 877–905, 2008.

[181] A. Repetti and Y. Wiaux, "A forward-backward algorithm for reweighted procedures: Application to radio-astronomical imaging," in *IEEE ICASSP*, pp. 1434–1438, 2020.

[182] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier analysis and applications*, vol. 14, pp. 877–905, 2008.

[183] A. Onose, A. Dabbech, and Y. Wiaux, "An accelerated splitting algorithm for radio-interferometric imaging: when natural and uniform weighting meet," *Monthly Notices of the Royal Astronomical Society*, vol. 469, no. 1, pp. 938–949, 2017.

[184] P.-A. Thouvenin, A. Abdulaziz, A. Dabbech, A. Repetti, and Y. Wiaux, "Parallel faceted imaging in radio interferometry via proximal splitting (Faceted HyperSARA): I. Algorithm and simulations," *Monthly Notices of the Royal Astronomical Society*, vol. 521, no. 1, pp. 1–19, 2023.

[185] S. V. Kartik, R. E. Carrillo, J.-P. Thiran, and Y. Wiaux, "A Fourier dimensionality reduction model for big data interferometric imaging," *Monthly Notices of the Royal Astronomical Society*, vol. 468, no. 2, pp. 2382–2400, 2017.

[186] X. Cai, L. Pratley, and J. D. McEwen, "Online radio interferometric imaging: assimilating and discarding visibilities on arrival," *Monthly Notices of the Royal Astronomical Society*, vol. 485, no. 4, pp. 4559–4572, 2019.

[187] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.

[188] J. Jonas and MeerKAT Team, "The MeerKAT Radio Telescope," in *MeerKAT Science: On the Pathway to the SKA*, p. 1, 2016.

[189] A. Aghabiglou, C. Chu, A. Dabbech, and Y. Wiaux, "The R2D2 deep neural network series for fast high-dynamic range imaging in radio astronomy," *Astrophysical Journal*, 2023.

[190] H. Le, N. Gillis, and P. Patrinos, "Inertial block proximal methods for non-convex non-smooth optimization," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *Proceedings of Machine Learning Research*, pp. 5671–5681, PMLR, 13–18 Jul 2020.

[191] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[192] S. J. Wright, "Coordinate descent algorithms," *Mathematical programming*, vol. 151, no. 1, pp. 3–34, 2015.

[193] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke, "Coordinate descent converges faster with the gauss-southwell rule than random selection," in *Proc. ICML'15*, 2015.

[194] S. Salzo and S. Villa, "Parallel random block-coordinate forward–backward algorithm: a unified convergence analysis," *Mathematical Programming*, vol. 193, pp. 225–269, May 2022.

[195] L. Briceño-Arias, J. Deride, and C. Vega, "Random activations in primal-dual splittings for monotone inclusions with a priori information," *J Optim Theory Appl*, vol. 192, pp. 56–81, 2022.

[196] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 1221–1248, 2015.

[197] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, pp. 1–38, 2014.

[198] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, vol. 156, pp. 433–484, 2016.

[199] O. Fercoq and P. Richtárik, "Accelerated, parallel, and proximal coordinate descent," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 1997–2023, 2015.

[200] S. Cadoni, E. Chouzenoux, J.-C. Pesquet, and C. Chaux, "A block parallel majorize-minimize memory gradient algorithm," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3194–3198, IEEE, 2016.

[201] H. Namkoong, A. Sinha, S. Yadlowsky, and J. C. Duchi, "Adaptive sampling probabilities for non-smooth optimization," in *International Conference on Machine Learning*, pp. 2574–2583, PMLR, 2017.

[202] C.-P. Lee and S. J. Wright, "Random permutations fix a worst case for cyclic coordinate descent," *IMA Journal of Numerical Analysis*, vol. 39, pp. 1246–1275, July 2019.

[203] R. Sun and Y. Ye, "Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version," *Mathematical Programming*, vol. 185, pp. 487–520, 2021.

[204] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, pp. 475–494, 2001.

[205] X. Li, A. Milzarek, and J. Qiu, "Convergence of random reshuffling under the Kurdyka-Łojasiewicz inequality," *SIAM Journal on Optimization*, vol. 33, no. 2, pp. 1092–1120, 2023.

[206] L. Zheng, E. Riccietti, and R. Gribonval, "Efficient identification of butterfly sparse matrix factorizations," *SIAM Journal on Mathematics of Data Science*, vol. 5, no. 1, pp. 22–49, 2023.

[207] F. Abboud, É. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli, "An alternating proximal approach for blind video deconvolution," *Signal Processing: Image Communication*, vol. 70, pp. 21–36, 2019.

[208] Y. Xu and W. Yin, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *J Sci Comput*, vol. 72, pp. 700–734, 2017.

[209] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[210] Z. Qu, P. Richtárik, and T. Zhang, "Quartz: Randomized dual coordinate ascent with arbitrary sampling," *Advances in neural information processing systems*, vol. 28, 2015.

[211] A. Chambolle, M. J. Ehrhardt, P. Richtárik, and C.-B. Schonlieb, "Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications," *SIAM Journal on Optimization*, vol. 28, no. 4, pp. 2783–2808, 2018.

[212] H. Attouch and J. Bolte, "On the convergence of the proximal algorithm for nonsmooth functions involving analytic features," *Mathematical Programming*, vol. 116, no. 1-2, pp. 5–16, 2009.

[213] 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*, vol. 35, no. 2, pp. 438–457, 2010.

[214] J. Bolte, A. Daniilidis, O. Ley, and L. Mazet, "Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity," *Trans. Am. Math. Soc.*, vol. 362, no. 6, pp. 3319–3363, 2010.

[215] 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*, vol. 137, no. 1, pp. 91–129, 2013.

[216] D. Noll, "Convergence of non-smooth descent methods using the Kurdyka-Łojasiewicz inequality," *Journal of Optimization Theory and Applications*, vol. 160, pp. 553–572, 2014.

[217] 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*, vol. 18, no. 5, pp. 1199–1232, 2018.

[218] P. Abry and P. Flandrin, "On the initialization of the discrete wavelet transform algorithm," *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 32–34, 1994.

[219] P. Abry, P. Goncalves, and J. L. Véhel, *Scaling, fractals and wavelets*. John Wiley & Sons, 2013.

[220] A. Gonon, L. Zheng, P. Carrivain, and Q.-T. Le, "Make Inference Faster: Efficient GPU Memory Management for Butterfly Sparse Matrix Multiplication," *arXiv preprint arXiv:2405.15013*, 2024.

[221] HPCwire, "AWS Upgrades its GPU-Backed AI Inference Platform," Mar 2019. Accessed: April 2024.

[222] J. Barr, "Amazon EC2 Update – Inf1 Instances with AWS Inferentia Chips for High Performance Cost-Effective Inferencing," 2019. Accessed: April 2024.

[223] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, G. Moser, A. De Giorgi, L. Fang, Y. Chen, M. Chi, *et al.*, "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning," *IEEE geoscience and remote sensing magazine*, vol. 6, no. 3, pp. 10–43, 2018.

[224] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279–317, 2019.

[225] C. Castera, *Inertial and Second-order Optimization Algorithms for Training Neural Networks*. PhD thesis, Institut National Polytechnique de Toulouse-INPT, 2021.

[226] J. D. Lee, Y. Sun, and M. A. Saunders, "Proximal newton-type methods for minimizing composite functions," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1420–1443, 2014.

[227] L. Gaedke-Merzhäuser, A. Kopaničáková, and R. Krause, "Multilevel minimization for deep residual networks," *ESAIM: Proceedings and Surveys*, vol. 71, pp. 131–144, 2021.

[228] C. Ponce, R. Li, C. Mao, and P. Vassilevski, "Multilevel-in-width training for deep neural network regression," *Numerical Linear Algebra with Applications*, vol. 30, no. 5, p. e2501, 2023.

[229] E. Treister, J. S. Turek, and I. Yavneh, "A multilevel framework for sparse optimization with application to inverse covariance estimation and logistic regression," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. S566–S592, 2016.

[230] R. H. Chan and K. Chen, "A multilevel algorithm for simultaneously denoising and deblurring images," *SIAM Journal on Scientific Computing*, vol. 32, no. 2, pp. 1043–1063, 2010.

[231] T. F. Chan and K. Chen, "An optimization-based multilevel algorithm for total variation image denoising," *Multiscale Modeling & Simulation*, vol. 5, no. 2, pp. 615–645, 2006.

[232] A. Chambolle and R. Tovey, "FISTA in Banach spaces with adaptive discretisations," *Computational Optimization and Applications*, vol. 83, no. 3, pp. 845–892, 2022.

[233] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 146, pp. 37–75, 2014.

[234] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[235] J. Bolte, C. W. Combettes, and E. Pauwels, "The iterates of the Frank–Wolfe algorithm may not converge," *Mathematics of Operations Research*, 2023.

[236] M. Fornasier and C.-B. Schönlieb, "Subspace Correction Methods for Total Variation and ℓ_1-Minimization," *SIAM Journal on Numerical Analysis*, vol. 47, pp. 3397–3428, Jan. 2009.

[237] M. Fornasier, Y. Kim, A. Langer, and C.-B. Schönlieb, "Wavelet Decomposition Method for L_2-TV-Image Deblurring," *SIAM Journal on Imaging Sciences*, vol. 5, pp. 857–885, Jan. 2012.

[238] C. Vonesch and M. Unser, "A Fast Multilevel Algorithm for Wavelet-Regularized Image Restoration," *IEEE Transactions on Image Processing*, vol. 18, pp. 509–523, Mar. 2009.

Remerciements

J'écris ces remerciements bien après la soutenance. Parce qu'elle s'est si bien passée. Parce que je veux pouvoir remercier à leur juste valeur celles et ceux qui m'ont permis d'en arriver là. Les émotions qui m'ont traversées à la fin, ont été l'expression directe de ma gratitude envers les personnes qui m'ont entouré, accompagné, et soutenu tout au long de ces trois années de thèse à l'ENS.

Pour commencer, je veux remercier mes trois encadrant et encadrantes de thèse, Paulo, Nelly et Elisa. Mes recherches pour trouver une thèse ont été chaotiques, et pour je ne sais quelle raison, j'ai su immédiatement lorsque nous avons discuté pour la première fois que vous étiez les trois personnes avec qui je devais faire ma thèse. À tel point que j'ai appelé un samedi Elisa - qui a mis son numéro sur son site - après les circonvolutions dont j'ai l'habitude pour pouvoir confirmer mon choix. Et c'est allé au-delà de toutes mes espérances, du premier au dernier jour de ces trois ans.

Tout d'abord par votre soutien scientifique. Vos trois points de vue, vos trois visions m'ont permis en tout instant de compléter autant que possible mes recherches et mon travail, qui sont partis de connaissances frémissantes à une maîtrise de ce qu'il m'a été nécessaire pour achever cette thèse.

Avant tout parce que grâce à vous, ces trois années ont été formidables, et le mot est faible ici. Je crois que je n'aurais pas pu accomplir tout le travail qu'a représenté cette thèse si je n'avais pas su que je pouvais compter sur vous pour que je puisse tirer le maximum de chaque jour. J'ai fait beaucoup de pas de côté, volontaires pour certains - notamment mes choix « artistiques » -, involontaires pour d'autres, mais vous avez toujours su me guider dans une direction qui est la bonne.

Ce n'est pas tous les jours que l'on peut être soi sans en douter, et je vous en dois beaucoup. Mille mercis.

Je veux aussi remercier les membres du jury. Merci beaucoup Aude Rondepierre d'avoir accepté de rapporter ma thèse, et d'y avoir accordé plus d'attention que je ne pouvais espérer. Thanks a lot Ivan Selesnick for also accepting to report my thesis. I was also delighted that Panos Parpas, whose work helped me tremendously at the beginning of my thesis, accepted to be part of this jury! It was a pleasure meeting you after these three years. Merci beaucoup Bruno (en compagnie de Bora) de m'avoir suivi épisodiquement durant cette thèse, et d'avoir été là pour cette conclusion ! Enfin je veux remercier Jean-Christophe Pesquet d'avoir accepté de présider ce jury, c'était un honneur pour moi.

J'ai eu l'occasion pendant ma thèse d'entamer plusieurs collaborations qui ont non seulement enrichi ma thèse scientifiquement mais aussi humainement. Merci Audrey - de m'avoir fait confiance - et Yves de m'avoir permis de travailler sur la radio-interférométrie, et aussi de m'avoir fait découvrir Edinburgh ! Merci aussi Luis, pour tes remarques et suggestions précieuses, qui nous ont mené à ce dernier chapitre, chapitre dont je suis le plus fier sur le plan théorique.

Merci aussi à tous les membres permanents de l'équipe, Rémi G, tes conseils et ton entrain m'ont toujours été très précieux; Mathurin, pour ton soutien indéfectible contre vents, énergies, et marées; Titouan, pour ton humour et tes bonnes paroles quotidiennes ; Pascal, pour ta bonne humeur, tes blagues que je suis rapide à comprendre mais qu'il faut m'expliquer longtemps, et tes vraies fausses informations ; Marion et Simon, pour les bons moments et discussions qu'on a partagés. Merci à vous.

Cette thèse je l'ai commencée avec six autres doctorants désormais docteurs, Clément, Samir, Anthony, Tung, Léon, Antoine, que j'ai apprécié dès le premier jour. Vous m'avez inspiré à donner le meilleur de moi-même. Vous avez aussi fait de cette thèse, démarré comme une aventure solitaire, un travail d'équipe. Pour ça, je ne pourrai vous remercier assez !

Merci aussi à toutes celles et ceux que j'ai croisé au cours de ces trois années à l'ENS et ailleurs ! À tous les anciens et affiliés de Dante et Ockham, Sybille, Wassim, Rémi V, Ayoub, Badr, Meriem, Esther, Luc, Solène, Valérie, merci beaucoup ! J'ai eu grand plaisir à vous côtoyer, et j'en aurai encore plus si on se recroise ! Merci Clara, merci Gabriel, votre bonne humeur et vos anecdotes outre-atlantiques vont me manquer. Merci Marie pour ces conférences qu'on a partagées, et à ta family ! Merci beaucoup Myriam, nos sessions hebdomadaires d'escalades m'ont et vont me manquer.

Aux collègues du laboratoire de Physique, Léo, Victor, Nils, Juliana, Guillermo, Julian, merci à vous, ce fut un plaisir ! Merci aussi aux MALIP, au CBP, à Patrice, à Diane, et aux autres résidents du M7-1H, ce couloir va bien me manquer !

Merci à vous tous pour tout ce qu'on a partagé à Lyon en et hors du bureau

Merci beaucoup aux nouveaux Arthur, Maël et mon « remplaçant » Edgar. L'ambiance de l'équipe ces derniers mois n'auraient pas été la même sans vous ! Je vous souhaite de vivre votre thèse à fond, et de la terminer dans la même joie que celle que j'ai vécue.

Enfin je veux remercier mes camarades de bureau de ces quasi deux dernières années, Anne, Can et Étienne.

Merci beaucoup Étienne, tu as mis le rythme de chaque journée par nos discussions, nos cafés. Charles Pasco t'en doit une.

Merci beaucoup Can, tu es une des plus belles personnes que je connaisse, et ce fut un vrai plaisir de faire un bout de chemin avec toi !

Merci beaucoup Anne, sans toi cette dernière année n'aurait pas été pareille dans son déroulement et son dénouement.

Avec vous tous, mon Rambouillet est au complet.

Pour terminer, le plus grand des mercis je le réserve à mes amis, et famille, qui depuis tant d'années sont à mes côtés.

Abstract

The size of image restoration problems is constantly increasing. This growth poses a major scaling problem for optimization algorithms, which struggle to provide satisfactory solutions in a reasonable amount of time.

Among the methods proposed to overcome this challenge, multilevel methods seem to be an ideal candidate. By systematically reducing the size of the problem, the computational cost of solving it can be drastically decreased. This type of approach is standard in the numerical solution of partial differential equations (PDEs), with theoretical guarantees and practical demonstrations to explain their success.

However, current multilevel optimization methods do not have the same guarantees nor the same performance. In this thesis, we propose to bridge a part of this gap by introducing a new multilevel algorithm, IML FISTA, which has the optimal theoretical convergence guarantees for convex non-smooth optimization problems, i.e. convergence to a minimiser and convergence rate of the objective function to a minimum value. IML FISTA is also able to handle state-of-the-art regularizations in image restoration.

By comparing IML FISTA with standard algorithms on many image restoration problems: deblurring, denoising, reconstruction of missing pixels for colour and hyperspectral images, and reconstruction of radio-interferometric images, we show that IML FISTA is capable of significantly speeding up the resolution of these problems. As IML FISTA's framework is sufficiently general, it can be adapted to many other image restoration problems.

We conclude this thesis by proposing a new point of view on multilevel algorithms, by demonstrating their equivalence, in certain cases, with coordinate descent algorithms, which are much more widely studied in the non-smooth optimization literature. This new theoretical framework allows us to analyse multilevel algorithms more rigorously, and in particular to extend their convergence guarantees to non-smooth and non-convex problems. This framework is less general than that of IML FISTA, but it paves the way for a more theoretically robust design of multilevel algorithms.

Résumé

La taille des problèmes de restauration d'images ne fait qu'augmenter. Cette croissance pose un problème majeur de passage à l'échelle pour les algorithmes d'optimisation, qui peinent à fournir des solutions satisfaisantes en un temps raisonnable.

Parmi les méthodes proposées pour surmonter ce défi, les méthodes multi-niveaux semblent être un candidat idéal. En réduisant de manière systématique la dimension du problème, le coût computationnel nécessaire à sa résolution peut diminuer drastiquement. Ce type d'approche est classique pour la résolution numérique des équations aux dérivées partielles (EDP), avec des garanties théoriques et des démonstrations pratiques pour expliquer leur succès.

Cependant, les méthodes actuelles d'optimisation multi-niveaux n'ont pas les mêmes garanties, ni les mêmes performances. Dans cette thèse, nous proposons de combler une partie de cet écart en introduisant un nouvel algorithme multi-niveau, IML FISTA, possédant les garanties de convergence théoriques optimales pour les problèmes d'optimisation convexes non-lisses, i.e., convergence vers un minimiseur et taux de convergence de la fonction objectif vers une valeur minimale. IML FISTA est aussi en mesure de traiter les régularisations de l'état-de-l'art en restauration d'images.

En comparant IML FISTA aux algorithmes standards sur un grand nombre de problèmes de restauration d'images: défloutage, débruitage, reconstruction de pixels manquants pour des images en couleur et des images hyperspectrales, ainsi qu'en reconstruction d'images radio-interférométriques, nous montrons qu'IML FISTA est capable d'accélérer la résolution de ces problèmes de manière significative. Le cadre d'IML FISTA est suffisamment général pour s'adapter à de nombreux autres problèmes de restauration d'images.

Nous concluons cette thèse en proposant un nouveau point de vue sur les algorithmes multi-niveaux, en démontrant leur équivalence, dans certains cas, avec les algorithmes de descente par coordonnées qui sont nettement plus étudiés dans la littérature de l'optimisation non-lisse. Ce nouveau cadre théorique nous permet d'analyser les algorithmes multi-niveaux de manière plus rigoureuse, et notamment d'étendre leurs garanties de convergence à des problèmes non-lisses et non-convexes. Ce cadre est moins général que celui d'IML FISTA, mais il ouvre la voie à une conception plus solide sur le plan théorique des algorithmes multi-niveaux.