JOHANNES KEPLER UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche Fakultät

Subspace Correction and Domain Decomposition Methods for Total Variation Minimization

DISSERTATION

zur Erlangung des akademischen Grades

Doktor

im Doktoratsstudium der

Technischen Wissenschaften

Eingereicht von:
Dipl.-Ing. Andreas Langer

Angefertigt am:
Johann Radon Institute for Computational and Applied Mathematics

Beurteilung:
Prof. Dr. Massimo Fornasier, Johann Radon Institute for Computational and Applied Mathematics (Betreuung)
Prof. Dr. Daniel Cremers, Technische Universität München

Linz, Juli 2011

To my mother, Maria Langer

Abstract

The minimization of the total variation is a well-established technique of preserving edges and discontinuities. Therefore it has wide applicability in image processing, where one is interested in recovering a corrupted image while preserving edges in the image. In the last decades, in the literature, there have been introduced many different approaches and algorithms for minimizing the total variation. These standard techniques are iterative-sequentially formulated and therefore not able to solve large scale simulations in acceptable computational time. For such large problems we need to address methods that allow us to reduce the problem to a finite sequence of subproblems of a more manageable size, perhaps computed by one of the standard techniques. With this aim, we introduce subspace correction and domain decomposition methods for total variation minimization. Such methods split the space of the initial problem into several smaller subspaces. By restricting the function to be minimized to the subspaces, a sequence of local problems, which may be solved easier and faster than the original problem, is constituted. Then the solution of the initial problem is obtained via the solutions of the local subproblems by "gluing" them together. In the case of domain decomposition for total variation minimization the crucial difficulty is the correct treatment of the interfaces of the domain decomposition patches, with the preservation of crossing discontinuities and the correct matching where the solution is continuous instead. Due to the fact that the total variation is non-smooth and non-additive, one encounters additional difficulties in showing convergence of more general subspace correction strategies to global minimizers. We show a classical counterexample from the literature, which emphasizes that for non-smooth and non-additive problems such alternating techniques are far from being obviously converging to an expected minimizer.

Moreover we discuss a subspace correction method for total variation minimization based on an orthogonal wavelet decomposition. For this algorithm we are able to construct another counterexample, which confirms that in general we cannot expect convergence of such splitting algorithms to a minimizer of the original problem.

Nevertheless, we are able to propose an implementation of overlapping and non-overlapping domain decomposition algorithms for total variation minimization with the guarantee of convergence to a minimizer of the original functional and the monotonic decay of the energy. Let us stress that these are the first successful attempts of addressing domain decomposition strategies for the non-linear, non-additive, and non-smooth problem of total variation minimization with a rigorous convergence analysis. We provide several numerical experiments, showing the successful application of the algorithm for the restoration of 1D signals and 2D images.

Zusammenfassung

Die Minimierung der totalen Variation ist eine gängige Technik, Kanten und Unstetigkeiten zu erhalten. Deshalb genießt sie weite Anwendung in der Bildverarbeitung, wo man daran interessiert ist, beschädigte Bilder zu rekonstruieren, wobei Kanten in den Bildern erhalten bleiben sollen. In den letzten Jahrzehnten wurden in der Literatur viele verschiedene Herangehensweisen und Algorithmen zur Minimierung der totalen Variation vorgestellt. Diese Standardtechniken sind iterativ-sequenziell formuliert und deshalb nicht in der Lage, sehr große Simulationen in akzeptabler Berechnungszeit zu lösen. Für solche großen Probleme müssen wir Methoden anwenden, welche uns erlauben das ursprüngliche Problem zu einer endlichen Folge von Teilproblemen von leichter handhabbarer Größe zu reduzieren, die dann womöglich durch eine der Standardtechniken gelöst wird. Mit dieser Absicht führen wir Unterraum-Korrekturverfahren und Gebietszerlegungsmethoden zur Minimierung der totalen Variation ein. Solche Methoden teilen den Raum des Anfangsproblems in mehrere kleinere Teilräume. Durch die Beschränkung der zu minimierenden Funktion auf die Unterräume wird eine Folge von lokalen Problemen, welche leichter und schneller als das originale Problem gelöst werden können, gebildet. Die Lösung des ursprünglichen Problems wird dann durch „zusammenkleben“ der Lösungen der lokalen Teilprobleme erreicht. Im Fall einer Gebietszerlegung ist die wesentliche Schwierigkeit bei der Minimierung der totalen Variation die korrekte Behandlung der Schnittstellen zwischen den Teilgebieten mit der Erhaltung von querenden Unstetigkeiten und Stetigkeiten. Auf Grund der Tatsache, dass die totale Variation nicht glatt und nicht additiv ist, begegnen wir zusätzlichen Schwierigkeiten, Konvergenz von allgemeinen Unterraum-Korrekturverfahren zu globalen Minimierern zu beweisen. Wir zeigen ein klassisches Gegenbeispiel aus der Literatur, welches uns verdeutlicht, dass für nicht glatte und nicht additive Probleme solche alternierenden Techniken im Allgemeinen nicht zum erwarteten Minimierer konvergieren.

Weiters stellen wir ein Unterraum-Korrekturverfahren zur Minimierung der totalen Variation vor, welches auf einer orthogonalen Wavelet-Zerlegung basiert. Für diese spezielle Methode sind wir in der Lage, ein weiteres Gegenbeispiel zu konstruieren, welches bestärkt, dass wir generell für solche Algorithmen keine Konvergenz zu einem Minimierer des originalen Problems erwarten können.

Trotzdem ist es uns möglich, für die Minimierung der totalen Variation eine Implementierung von überlappenden und nicht überlappenden Gebietszerlegungsalgorithmen vorzuschlagen, die die Konvergenz zu einem Minimierer des originalen Funktionals und den monotonen Abfall der Energie garantiert. Wir betonen, dass dies die ersten erfolgreichen Versuche sind, Gebietszerlegungsstrategien für das nicht lineare, nicht additive und nicht glatte Problem der Minimierung der totalen Variation, deren Konvergenz ausführlich analysiert ist, anzuwenden. Wir zeigen numerische Experimente, die die erfolgreiche Anwendung dieser Algorithmen unterstreichen.

Acknowledgments

I would like to express my deep gratitude to my supervisor Massimo Fornasier for introducing me to interesting and challenging problems, for encouraging me to believe in myself, for giving me and showing me great new opportunities, and for being a mentor and a friend. Moreover I am very thankful to Daniel Cremers for nicely agreeing to be the second referee of this thesis.

Many thanks go to Luminiţa Vese for her great support, for giving me the possibility to visit the Department of Mathematics at the University of California, Los Angeles, and providing financial support during my stay there. Moreover, she helped me to make my visit very enjoyable and a very positive experience. It was an honor for me to visit and to work at the University of California, Los Angeles, and I am glad that I had the chance to meet and to discuss with several very interesting mathematicians there. I had a great time!

I would like to thank all my collaborators for giving me new insights in mathematical problems and for everything I learnt from them. In particular, I owe many thanks to Carola-Bibiane Schönlieb for encouraging and inspiring me, and for being a good colleague and friend. Working with you is great fun! I am also particularly thankful to Yunho Kim and Stanley Osher for interesting and fruitful discussions, which enlarged my scope tremendously.

A lot of thanks go to all my colleagues at the Johann Radon Institute for Computational and Applied Mathematics (RICAM) and particularly to the group “Analysis of Partial Differential Equations” for their hospitality and for providing an excellent scientific environment. In particular, I would like to thank Francesco Solombrino for his help, hints, and support that improved this thesis immensely.

I gratefully acknowledge the financial support provided by the FWF project Y 432-N15 START-Preis *Sparse Approximation and Optimization in High Dimensions* and the project WWTF Five senses-Call 2006, *Mathematical Methods for Image Analysis and Processing in the Visual Arts.*

I owe very special thanks to my family, in particular to my parents Maria and Thomas Langer and my brother Gregor Langer, who supported me through the years in good and in bad times, and to my girlfriend, Mădălina Hodorog, for always being there for me, for supporting me, and for encouraging me. I also would like to thank all my friends for distracting me from work in my spare time.

Contents

Chapter 1

Introduction

1.1 Total Variation Minimization

In applications in image processing, one wants to recover at best an observed image, which is corrupted by a measurement device and additionally by some noise e. Then the observed data g can be described as

$$g = Tu + e, \tag{1.1}$$

where u is the unknown image and T a linear operator modeling the image-formation device. We are in particular interested in the recovery of u from the given noisy observed image g, when the operator T is not invertible or ill-conditioned. This is indeed an inverse problem and thus regularization techniques are required to restore the unknown image [55]. A good approximation of u may be obtained by minimizing a functional of the type

$$functional = data\ term + regularization\ term.$$

The *data term* enforces the consistency between the recovered and measured signal, while the *regularization term* prevents overfitting. In our case we obtain then the following minimization problem

$$\arg\min_u \int_\Omega |Tu - g|^2 dx + 2\alpha F(u), \tag{1.2}$$

where $\Omega \subset \mathbb{R}^2$ is the image domain, F is any functional representing the regularization term, and $\alpha > 0$ is a fixed regularization parameter weighting the importance of the two terms.

We wonder now what will be a good choice for the regularization term F. If we use a standard Tikhonov regularization we will consider, for example, the function $F(u) = \int_\Omega |\nabla u|^2 dx$, where $\nabla u(x) = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}\right)$ is the gradient of u in $x \in \Omega$. Then the function space for which both the data term and the regularization term are well-defined is

$$W^{1,2}(\Omega) = \{u \in L^2(\Omega) \ : \ \nabla u \in [L^2(\Omega)]^2\},$$

i.e., the Sobolev space of L^2-functions with L^2-derivatives. This choice of regularization may have advantages, since the corresponding problem to solve is linear. However, using the L^2-norm of the gradient as a regularization term allows us to remove noises but the minimization of (1.2) performs too much smoothing and hence does not preserve edges (i.e. intensity jumps along curves) and discontinuities across hypersurfaces, i.e., across lines in 2-dimensions, see Figure 1.1. For our purposes a good regularization term should ensure some smoothing and should preserve edges and discontinuities.

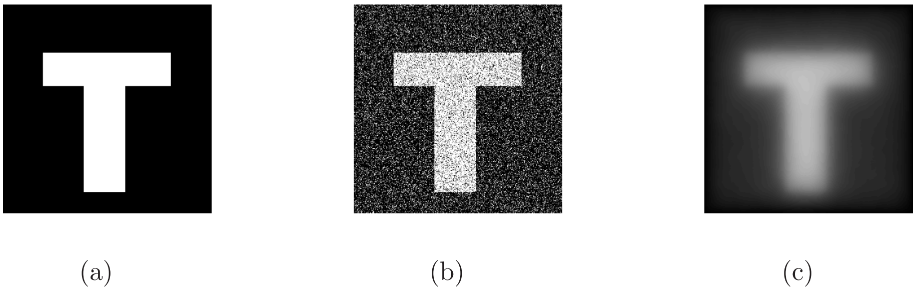

(a) (b) (c)

Figure 1.1: *The left picture shows the original image, which is then corrupted by some noise, see (b). The right picture depict the restoration with* $F(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 dx$.

A possible solution was given by Geman and Geman [68], who described a stochastic relaxation algorithm, which generates a sequence of images that converges in an appropriate sense to the maximum a posteriori estimate. More precisely, they considered a discrete image $g = (g_{i,j})_{1\leq i,j\leq N}$ for $N \in \mathbb{N}$ and introduced an auxiliary variable $\ell = (\ell_{i+1/2,j}, \ell_{i,j+1/2})_{i,j}$ to detect edges in the picture, i.e., $\ell_{i+1/2,j} = 1$ if there is an edge between (i,j) and $(i+1,j)$ and 0 otherwise. In the case when e is some Gaussian noise with mean 0 and standard deviation σ, the reconstruction of the observed image g, modeled as in (1.1), is then obtained by minimizing the following energy with respect to u and ℓ

$$\mathbb{P}(u,\ell) + \frac{1}{2\sigma^2}\sum_{i,j} |g_{i,j} - Tu_{i,j}|^2$$

where $\mathbb{P}$ is an a priori probability density which might be of the form

$$\mathbb{P}(u,\ell) = \lambda \sum_{i,j} \big((1-\ell_{i+1/2,j})(u_{i+1,j}-u_{i,j})^2 + (1-\ell_{i,j+1/2})(u_{i,j+1}-u_{i,j})^2\big) \\ + \mu \sum_{i,j} (\ell_{i+1/2,j} + \ell_{i,j+1/2})$$

with $\lambda, \mu > 0$ [25].

Mumford and Shah [89] formulated this approach in a continuous setting by observing that the set $\{\ell = 1\}$ can be interpreted as the edge collection $K \subset \Omega$, i.e., the set of 1-dimensional curves. Thus they introduced the following problem originally proposed for *image segmentation*

$$\min_{u,K} \lambda \int_{\Omega\setminus K} |\nabla u|^2 dx + \mu \text{length}(K) + \int_{\Omega} |u-g|^2 dx,$$

where $T = I$ (the identity operator). This functional generated a lot of interest in the last decades, see [2, 3, 40, 56, 65, 67, 82, 87].

In the context of *image restoration* Rudin, Osher and Fatemi [103] proposed to use the total variation as a regularization technique. We recall that for $u \in L^1_{loc}(\Omega)$

$$V(u,\Omega) := \sup \left\{ \int_{\Omega} u \operatorname{div} \varphi \; dx : \varphi \in \left[C^1_c(\Omega)\right]^2, \|\varphi\|_\infty \leq 1 \right\}$$

is the *variation* of u, where $C^1_c(\Omega)$ denotes the space of C^1-functions with compact support in Ω and $\|\varphi\|_\infty = \sup_x \sqrt{\sum_i \varphi_i^2(x)}$. Moreover, $u \in BV(\Omega)$, the *space of bounded variation functions* [2, 57], if and only if $V(u,\Omega) < \infty$. In this case, $|Du|(\Omega) = V(u,\Omega)$, where $|Du|(\Omega)$ is the total variation of the finite Radon measure Du, the derivative of u in the sense of distributions. If $u \in W^{1,1}$ (the Sobolev space of L^1-functions with L^1-distributional derivatives), then $|Du|(\Omega) = \int_\Omega |\nabla u| dx$. It is well-established that the total variation preserves edges and discontinuities across hypersurfaces [25, 33]. Additionally it is convex and therefore also the minimization problem, which reads then as

$$\arg\min_u \int_\Omega |Tu - g|^2 dx + 2\alpha |Du|(\Omega), \tag{1.3}$$

becomes convex. Hence many tools from convex optimization can be used to solve this problem. This is a big advantage with respect to the non-convex approach of Mumford and Shah, where the energy has to be minimized with respect to u and with respect to the edge collection K.

In this thesis we are interested in the efficient minimization in $BV(\Omega)$ of the functional

$$\mathcal{J}(u) := \|Tu - g\|^2_{L^2(\Omega)} + 2\alpha \left|Du\right|(\Omega), \tag{1.4}$$

where $T : L^2(\Omega) \to L^2(\Omega)$ is a bounded linear operator, $g \in L^2(\Omega)$ is a datum, and $\alpha > 0$ is a fixed regularization parameter. More precisely, we are concerned with minimizing $\mathcal{J}$ by means of *subspace correction* and *domain decomposition*. That means, instead of minimizing (1.4) on the whole $BV(\Omega)$ we split the space into several subspaces and minimize alternating on each subspace the functional of interest.

Below we describe a few relevant and established applications, where total variation minimization is already successfully applied. We are aware that such examples can be efficiently solved by several standard techniques proposed in the literature, see Section 1.1.2. However, these applications provide us with fundamental tests on which we can demonstrate that our proposed subspace correction and domain decompositon methods work efficiently. Let us emphasize that these methods are not restricted to such examples but can be used for more advanced applications, see below.

1.1.1 Applications

The minimization of the total variation is successfully used in many image processing problems, including image denoising, image deblurring, image inpainting, and image zooming [3, 33]. Besides applications in imaging, the total variation is also efficiently used in the computation of the evolution of hypersurfaces by mean curvature [24] as well as in problems in optimal control [22], for example, when a singular diffusion equation is involved [94]. In this thesis we will focus on problems which appear in image processing. In particular, we will show successful applications of the proposed algorithms for image restoration. Let us explain in short the most important applications in imaging successfully addressed by means of total variation minimization. We refer to the books [33, 104, 3], without being exhaustive, for a more detailed representation of these topics.

Image Denoising

Image denoising is the problem of removing the noisy part of an image. Typically random noise is modeled by a probability distribution, where the kind of distribution is chosen according to the application. In many cases the Gaussian distribution is used, while for tomography the Poisson distribution and for radar imaging (speckle noise) the Gamma distribution is required.

Is the corresponding noise of an observed image g additive with, for example, the underlying Gaussian distribution, then g can be represented by (1.1) where $T = I$ is the identity operator. This model of additive Gaussian noise is often used for testing the efficiency of denoising algorithms.

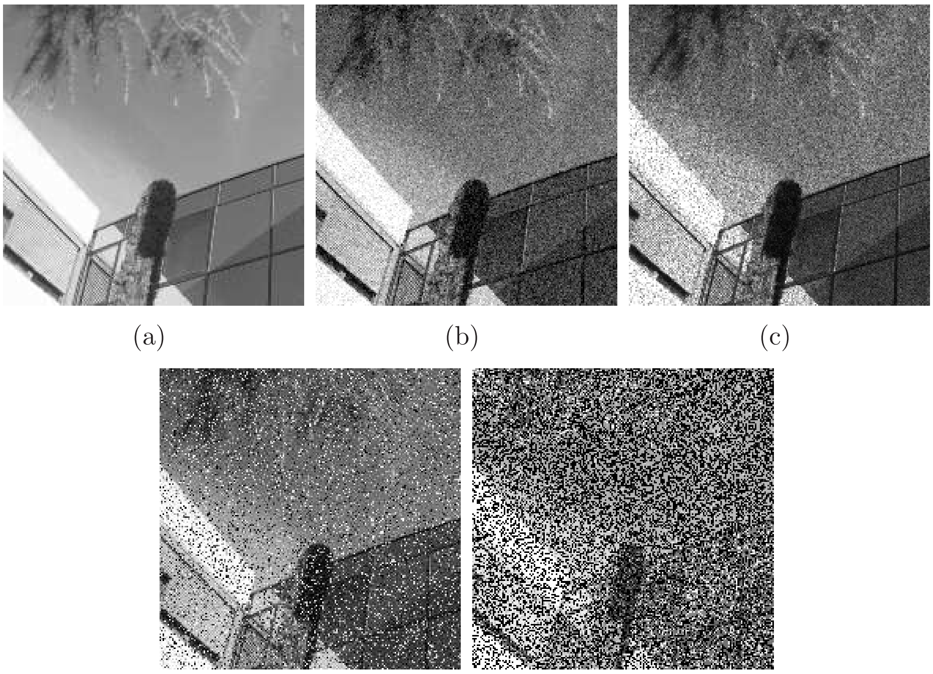
Another model arises if we assume that the noise e is acting multiplicatively on the unknown image u, i.e., $g = e \cdot u$. Then we speak about *multiplicative noise*. There exist even more sophisticated noise models, which describe the observed data by $g = e(u)$, where the noise e is nonlinear depending on u. A prominent example in this category is the *salt-and-pepper noise*, where the noisy discrete image $g = (g_{i,j})$ is given by

$$g_{i,j} = \begin{cases} c_{\min} & \text{with probability } p_1 \in [0,1] \\ c_{\max} & \text{with probability } p_2 \in [0,1] \\ u_{i,j} & \text{with probability } 1 - p_1 - p_2 \end{cases} ,$$

with $u = (u_{i,j})$, $c_{\min} = \min_{i,j}\{u_{i,j}\}$, $c_{\max} = \max_{i,j}\{u_{i,j}\}$, and $1 - p_1 - p_2 > 0$, see [28] for more details. In Figure 1.2 we show some typical examples of noisy images.

Noises in images are caused by many sources, for instance:

- in astronomical imaging: atmospheric inhomogeneity in terms of density, temperature, index of refraction, and so on;



(a) (b) (c)

(d) (e)

Figure 1.2: *(a) Original image corrupted by (b) additive Gaussian noise (c) multiplicative Gaussian noise (d) salt-and-pepper noise (e) speckle noise. Figures from [3].*

- in medical imaging: spontaneous motion and material inhomogeneity of tissues or organs;
- in night vision: fluctuation in heat, temperature, and infrared radiation;
- in general image acquisition: inherent thermal noises in electrooptical imaging devices, physical or chemical noises of the target systems to be imaged, and the inhomogeneity of intermediate media.

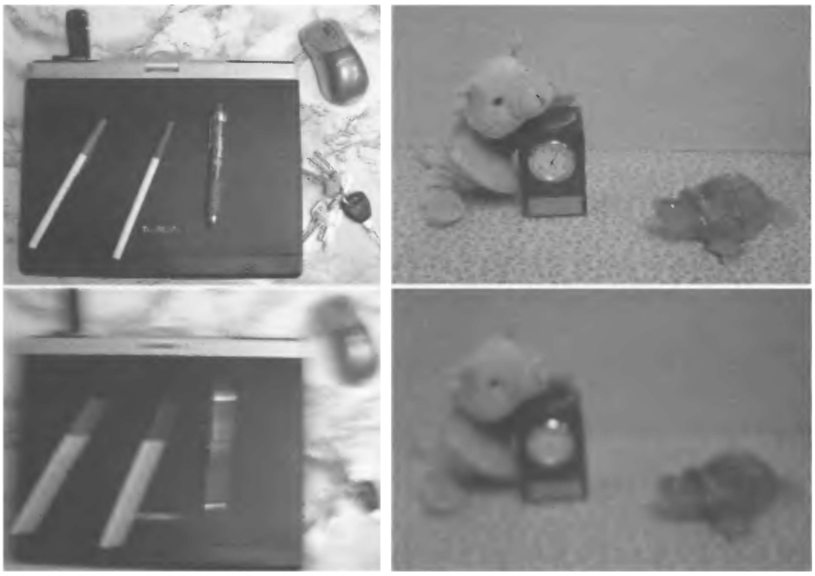
Looking at such pictures makes it often very difficult in spotting the important features and patterns. Such failures could be either clinically fatal (as in tumor detection) or scientifically costly [33].

Image Deblurring

The task of reconstructing a sharp image from its blurry observation is called *image deblurring*. This problem often arises in optical, medical, and astronomical applications, for example, in satellite imaging and remote sensing. There are three major categories of blurs (i) the out-of-focus blur, (ii) the motion blur, and (iii) the medium-induced blur, which are categorized according to their physical background, see [33] for more details. In Figure 1.3 we show examples of two different blurry images. However, in all the different cases a blurry image g is usually described as

$$g = Tu$$

where T is a blur operator modeled as a convolution $Tu = \kappa * u$, with kernel κ. If g is additionally also corrupted by some noise, the observed image data is then generally modeled as in (1.1).



(a) (b)

Figure 1.3: *(a) An example of motion blur (due to camera jittering); (b) an example of out-of-focus blur. Figures from [33].*

Image Inpainting

Image inpainting is the problem of filling in and recovering the missing parts of an image, which is essentially done by using the given information in the surroundings. Therefore it is a kind of interpolation.

Let the function g represent a partially corrupted image on the image domain Ω, with loss of information on a domain $D \subset \Omega$, cf. Figure 1.4. The domain D is called inpainting domain. The task of image inpainting is to reconstruct the image g given by

$$g = Tu$$

in the inpainting domain. Here now $T = 1_{\Omega \setminus D} \cdot$ is a multiplier, i.e., $Tu = 1_{\Omega \setminus D} \cdot u$, where $1_{\Omega \setminus D}$ is the characteristic function of $\Omega \setminus D$. If additionally additive noise is present in the observed data, then we generally describe g as in (1.1).

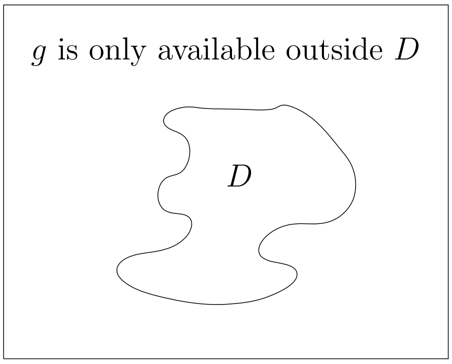


Figure 1.4: *The image inpainting task is to reconstruct the image in the missing part D.*

The term *inpainting* was originally used by museum restoration artists [54, 117] and introduced into digital image processing in the work of Bertalmio et al. [9], where a third order PDE inpainting model was constructed.

Image inpainting has a lot of applications in image processing, such as automatic scratch removal in digital photos and old films [9, 31], digital restoration of ancient paintings for conservation purposes [4], text erasing like dates, subtitles, or publicity [9, 5, 31, 32], special effects like object disappearance and wire removal for movie production [9, 32], zooming and super-resolution [5, 31, 83, 84, 112], lossy perceptual image coding [31], decomposition-based image interpolation [10, 114], landmark-based inpainting [73], removal of the laser dazzling effect [35], disocclusion [86, 93], and so on.

For a broad discussion on techniques for image inpainting, we refer to [105].

1.1.2 Methods for Total Variation Minimization

Several numerical strategies to perform efficiently total variation minimization have been proposed in the literature. Without claiming of being exhaustive, we list a few of the relevant methods.

Gradient descent approach
The most obvious way to tackle the problem of minimizing total variation constrained functionals is to derive and to solve the Euler-Lagrange equations. Rudin, Osher, and Fatemi [103] proposed to solve this nonlinear partial differential equation by adding an artificial time as an evolution parameter and by minimizing the energy along the gradient descent direction. Due to stability constraints in the time step size, this algorithm is very slow in convergence.

Linearized approach
Vogel et al. [115, 51] introduced a method using a *lagged diffusivity fixed point iteration* in order to solve the Euler-Lagrange equation of (1.3) with a "relaxed" total variation constraint. The same approach is described by Chambolle and Lions

in [27], where the algorithm is reinterpreted as a two step minimization as iterative re-weighted least squares. See also [43] for generalizations and refinements in the context of sparse recovery.

Primal-dual approach

The presence of a highly nonlinear and non-differentiable term in the Euler-Lagrange equation causes difficulties in convergence for Newton's method, even when combined with a globalization technique such as line search. In order to overcome these difficulties Chan et al. [29] proposed a primal-dual approach, which is based on the substitution of the non-differentiable term by a dual variable. Even more efficient algorithms in this category have been introduced recently in the literature [125, 99].

Projection onto convex sets

Chambolle suggested in [23] an iterative model based on projections onto convex sets by considering the dual formulation of the total variation and exploiting the corresponding optimality condition, see Section 3.2.6. Other mentionable work in this category is the forward-backward splitting method by Combettes and Wajs [39], who used a proximity operator, and the algorithm by Daubeschies et al. [45] that amounts to a projected Landweber iteration.

Bregman distance approach

The idea of the Bregman Iteration is to separate the total variation term and the L^2-term of (1.3) by a variable-splitting and then to solve a constrained optimization problem by penalizing and iteratively minimizing. Proposed by Osher et al. [96] as an iterative "add-back-noise" regularization procedure, it turned out that it is actually the well-known Augmented Lagrangian Method [72]. Many further variations on the theme of the Augmented Lagrangian Method and Bregman iterations have been proposed [19, 20, 69, 97, 122, 123, 124].

Graph cuts

Graph cut algorithms are established techniques in computer vision. Recently Darbon and Sigelle [46, 47] and Chambolle and Darbon [26] have introduced graph cut methods for the minimization of (1.3) for $T = I$ (the identity operator). This type of methods compute faster an exact solution in the sense of machine precision than all other iterative methods, known so far.

Approach by Nesterov

The approach presented by Nesterov [92] approximates the initial non-smooth total variation by a function with Lipschitz continuous gradient. Then a smooth function is minimized by an efficient gradient method of type [90, 91]. A modification of this algorithm is introduced by Weiss et al. [119].

These approaches differ significantly, and they provide a convincing view of the interest this problem has been able to generate and of its applicative impact. However, because of their iterative-sequential formulation, none of the mentioned methods is able to address in real-time, or at least in an acceptable computational time, extremely large problems, such as 4D imaging (spatial plus temporal dimensions) from functional magnetic-resonance in nuclear medical imaging, astronomical imaging or global terrestrial seismic tomography. Let us mention that with a clever implementation of these above mentioned standard techniques on a parallel architecture such as the graphics processing unit (GPU) one can accelerate them tremendously [98].

We are interested to address methods to such large scale simulations, which allow us to reduce the problem to a finite sequence of sub-problems of a more manageable size, perhaps computable by one of the methods listed above. Such methods are known under the name of *subspace corrections* and *domain decompositions*.

1.2 Basic Idea of Subspace Correction and Domain Decomposition

Subspace correction is a divide and conquer technique for solving partial differential equations by iteratively solving on each subspace an appropriate defined subproblem. Originally such methods were proposed for solving problems for which in the variational formulation a smooth energy is alternately minimized on each subspace. If the solution is non-smooth, then such splitting algorithms still work fine as long as the energy splits additively with respect to the subspace decomposition. By contrast for non-smooth and non-additive energies, such as (1.4), subspace correction methods are far from being obviously working successfully.

When analyzing such methods three main issues are of high interest: (i) convergence, (ii) rate of convergence, and (iii) the independence of the rate of convergence on the mesh size, which can be interpreted as a preconditioning strategy. For smooth energies these concerns are at large well-established, while for non-smooth energies convergence is ensured but no rate of convergence is usually known. In [116] Vonesch and Unser could provide preconditioning effects of a subspace correction algorithm for minimizing a non-smooth energy when applied to deblurring problems. However when dealing with the complicated case of a non-smooth and non-additive problem only very little is known. In this thesis we are now able to show for the non-smooth and non-additive case that decomposition strategies may converge to a minimizer of the original problem, cf. Theorem 4.2.8 and Theorem 4.3.1, and we are able to show preconditioning effects in certain cases. However, a complete description of the rate of convergence and independence of the mesh size is still a very open field of research.

In this section we introduce domain decomposition methods for smooth problems only, in order to describe the main ideas of such splitting techniques. Before we do so, let us describe shortly the importance of such methods.

The main reason for the success of subspace correction methods is the reduction of the dimension with a potential for parallelization. In particular, subspace correction is one of the most significant ways for devising parallel approaches that can benefit strongly from multiprocessor computers. Such parallel approaches are mandatory when one has to solve large-scale numerical problems, as they arise in many application of physics and engineering. Let us summarize the main advantages of such an approach, which include (i) dimension reduction; (ii) enhancement of parallelism; (iii) localized treatment of complex and irregular geometries, singularities and anomalous regions; (iv) and sometimes reduction of the computational complexity of the underlying solution method. There are a variety of iterative methods appearing in the literature that fall into the category of subspace correction methods, such as Jacobi method, Gauss-Seidel method, point or block relaxation methods, multigrid methods, and domain decomposition methods. These techniques can often be applied directly to the partial differential equation, but also the discretization of the problem is of major interest. We refer to [120] for more details on subspace correction methods.

The first known subspace correction strategy was proposed by H. A. Schwarz (1869)[106] who introduced an overlapping domain decomposition in order to prove the existence of harmonic functions on irregular regions that are the union of overlapping subregions [100, p 26]. Domain decomposition refers to the decomposition of the spatial domain into several subdomains. The original problem is then solved by iteratively solving alternating problems.

We focus now on domain decomposition methods and explain in more detail the underlying idea, which can be adapted to more general subspace correction methods. In particular, we review the non-overlapping domain decomposition as well as the alternating and parallel overlapping domain decomposition approaches in the case of a splitting of the physical domain into two subdomains. Their generalization to a partitioning into more domains requires more sophisticated techniques, such as coloring, see [30, 100, 111] for more details. For simplicity we discuss these methods now for the *Poisson problem*, i.e., second-order self-adjoint elliptic problem,

$$\mathcal{L}u \equiv -\Delta u = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega. \tag{1.5}$$

Here u is the unknown function, Δ denotes the Laplace operator, Ω is a 2-dimensional domain, i.e., $\Omega \subset \mathbb{R}^2$, with Lipschitz boundary $\partial\Omega$, and f is a given function. In what follows we summarize a few important results on domain decomposition techniques. For a broader discussion we refer to [100, 111].

1.2.1 Non-overlapping Domain Decomposition

Let us start by splitting the spatial domain Ω into two non-overlapping subdomains Ω_1 and Ω_2 such that $\overline{\Omega} = \overline{\Omega_1 \cup \Omega_2}$ and $\Omega_1 \cap \Omega_2 = \emptyset$, cf. Figure 1.5. We define the

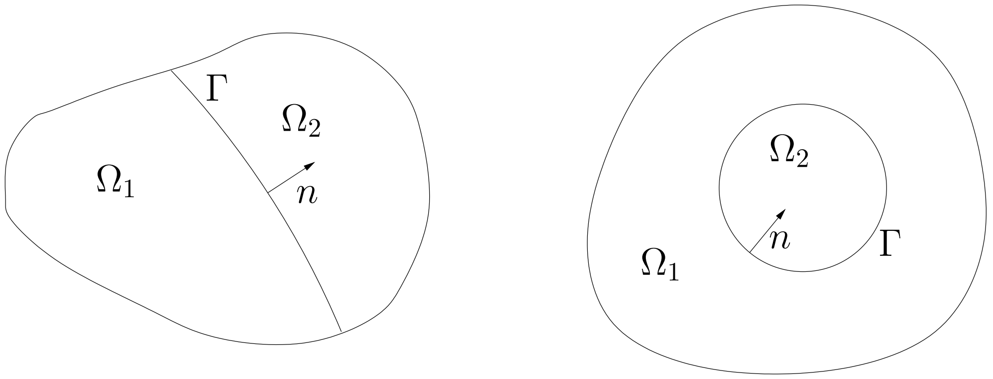


Figure 1.5: *Non-overlapping decomposition into two domains.*

interface between these two regions by $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$. In addition, we assume that the boundaries of the subdomains are Lipschitz continuous.

Then problem (1.5) can be formulated as

$$\begin{cases} \mathcal{L}u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1 = u_2 & \text{on } \Gamma \\ \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} & \text{on } \Gamma \\ \mathcal{L}u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega \end{cases}, \tag{1.6}$$

where each n is the outward pointed normal on Γ from Ω_1. Here we see that due to the partition of Ω the original problem (1.5) is replaced by two subproblems on each subdomain by imposing both Neumann and Dirichlet conditions on Γ. These conditions transmit information from one domain patch to the other and therefore they are called *transmission conditions*. The equivalence between the Poisson problem (1.5) and the multi-domain problem (1.6) is in general not obvious, but can be shown under suitable regularity assumptions on f, typically $f \in L^2(\Omega)$, by considering the associated variational formulation.

Iterative Methods

We will now focus on solving the multi-domain problem (1.6) by iterative methods. These methods typically introduce a sequence of subproblems on Ω_1 and Ω_2 for which Dirichlet or Neumann conditions at the internal boundary are provided, which play the role of the transmission conditions.

The Dirichlet-Neumann method For a given $\lambda^{(0)}$, solve for each $k \geq 0$

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f & \text{in } \Omega_1 \\ u_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma \\ u_1^{(k+1)} = \lambda^{(k)} & \text{on } \Gamma \end{cases} \text{ and } \begin{cases} \mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\ u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma \\ \frac{\partial u_2^{(k+1)}}{\partial n} = \frac{\partial u_1^{(k+1)}}{\partial n} & \text{on } \Gamma \end{cases} \tag{1.7}$$

with

$$\lambda^{(k+1)} := \theta u_{2|_\Gamma}^{(k+1)} + (1-\theta)\lambda^{(k)},$$

where $\theta > 0$ is an acceleration parameter.

We remark that this method does not necessarily converge, unless assumptions on the parameter θ or on Ω_1 and Ω_2 are made. However, if it is converging, then the rate of convergence is independent of the mesh size, see [85] for a convergence proof based on a functional analysis argument for partial differential equations.

The Neumann-Neumann method The Neumann-Neumann method proposed by Bourgat et al. [11] runs as follows: initialize $\lambda^{(0)}$ and solve for each $k \geq 0$

$$\begin{cases} \mathcal{L}u_i^{(k+1)} = f & \text{in } \Omega_i \\ u_i^{(k+1)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma \\ u_i^{(k+1)} = \lambda^{(k)} & \text{on } \Gamma \end{cases} \text{ and } \begin{cases} \mathcal{L}\psi_i^{(k+1)} = f & \text{in } \Omega_i \\ \psi_i^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma \\ \frac{\partial \psi_i^{(k+1)}}{\partial n} = \frac{\partial u_1^{(k+1)}}{\partial n} - \frac{\partial u_2^{(k+1)}}{\partial n} & \text{on } \Gamma \end{cases} \tag{1.8}$$

for $i = 1, 2$, with

$$\lambda^{(k+1)} := \lambda^{(k)} - \theta(\sigma_1 \psi_{1|_\Gamma}^{(k+1)} - \sigma_2 \psi_{2|_\Gamma}^{(k+1)}),$$

where $\theta > 0$ is an acceleration parameter and $\sigma_1, \sigma_2 > 0$ are averaging coefficients [100]. This method is ensured to converge and additionally in the case of a finite element discretization the rate of convergence is independent of the mesh size.

The Robin method Given $u_2^{(0)}$, then solve for $k \geq 0$

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f & \text{in } \Omega_1 \\ u_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma \\ \frac{\partial u_1^{(k+1)}}{\partial n} + \gamma_1 u_1^{(k+1)} = \frac{\partial u_2^{(k)}}{\partial n} + \gamma_1 u_2^{(k)} & \text{on } \Gamma \end{cases}$$

$$\text{and} \tag{1.9}$$

$$\begin{cases} \mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\ u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma \\ \frac{\partial u_2^{(k+1)}}{\partial n} - \gamma_2 u_2^{(k+1)} = \frac{\partial u_1^{(k+1)}}{\partial n} - \gamma_2 u_1^{(k+1)} & \text{on } \Gamma, \end{cases}$$

where γ_1 and γ_2 are non-negative acceleration parameters satisfying $\gamma_1 + \gamma_2 > 0$. The Robin method is ensured to converge [79], while we do not have any information about the rate of convergence nor estimates of the error reduction factor at each iteration.

The Method by Agoshkov and Lebedev The following non-overlapping domain decomposition algorithm was proposed by Agoshkov and Lebedev [1]: given $u_1^{(0)}$ and $u_2^{(0)}$, for each $k \geq 0$ we have to solve

$$
\begin{array}{ll}
\left\{\begin{array}{ll}
\mathcal{L}u_1^{(k+1/2)} = f & \text{in } \Omega_1 \\
u_1^{(k+1/2)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma \\
\frac{\partial u_1^{(k+1/2)}}{\partial n} + p_k u_1^{(k+1/2)} = \frac{\partial u_2^{(k)}}{\partial n} + p_k u_2^{(k)} & \text{on } \Gamma
\end{array}\right. & \\
u_1^{(k+1)} = u_1^{(k)} + \alpha_{k+1}(u_1^{(k+1/2)} - u_1^{(k)}) \qquad \text{in } \Omega_1 &
\end{array}
$$

and (1.10)

$$
\begin{array}{ll}
\left\{\begin{array}{ll}
\mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\
u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma \\
\frac{-q_k \partial u_2^{(k+1/2)}}{\partial n} + u_2^{(k+1/2)} = -q_k \frac{\partial u_1^{(k+1)}}{\partial n} + u_1^{(k+1)} & \text{on } \Gamma
\end{array}\right. & \\
u_2^{(k+1)} = u_2^{(k)} + \beta_{k+1}(u_2^{(k+1/2)} - u_1^{(k)}) \qquad \text{in } \Omega_2, &
\end{array}
$$

where $p_k, q_k \geq 0$ and $\alpha_{k+1}, \beta_{k+1} \in \mathbb{R}$ are free parameters. This algorithm is a generalization of many other methods, as the already mentioned Robin method (1.9), which is obtained by setting $p_k = \gamma_1$, $q_k = 1/\gamma_2$ and $\alpha_k = \beta_k = 1$ in (1.10). Similarly, one can also obtain the Dirichlet-Neumann method (1.7) by taking $p_k = q_k = 0$, $\alpha_k = \beta_k = 1$, and by noting that the roles of Ω_1 and Ω_2 are reversed.

Parallelism The Agoshkov-Lebedev method (1.10) and therefore also its special cases (1.7) and (1.9) are generating at each step two boundary value problems, the first in Ω_1 and the latter in Ω_2, to be solved sequentially. A simple modification frees these two subproblems from each other, which makes it more interesting in view of a parallel implementation. More precisely, when solving the boundary value problem in Ω_2 at the iteration step $k+1$, it is indeed enough to use $u_1^{(k)}$ instead of $u_1^{(k+1)}$. For example, in the Dirichlet-Neumann method (1.7) we simply replace the Neumann conditions on Γ by the new ones

$$\frac{\partial u_2^{(k+1)}}{\partial n} = \frac{\partial u_1^{(k)}}{\partial n} \quad \text{on } \Gamma.$$

1.2.2 Overlapping Domain Decomposition

In this section we describe the so-called *multiplicative* and *additive* Schwarz methods. Let us decompose the domain $\Omega \subset \mathbb{R}^2$ into two overlapping subdomains Ω_1 and Ω_2 such that $\Omega_1 \cap \Omega_2 \neq \emptyset$ and $\Omega = \Omega_1 \cup \Omega_2$, cf. Figure 1.6. Further we denote $\Gamma_1 = \partial\Omega_1 \cap \Omega_2$ and $\Gamma_2 = \partial\Omega_2 \cap \Omega_1$ the interior boundaries of the subdomains.

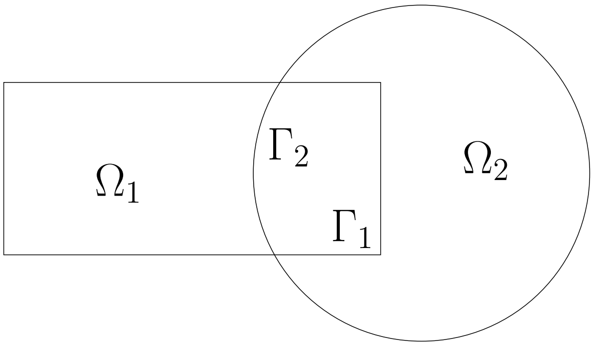


Figure 1.6: *Overlapping decomposition into two domains.*

Multiplicative Schwarz Method

The multiplicative Schwarz method starts with an initial value $u^{(0)}$ defined in Ω and vanishing on $\partial\Omega$ and computes a sequence of approximate solutions $u^{(1)}, u^{(2)}, \ldots$ by solving

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f & \text{in } \Omega_1 \\ u_1^{(k+1)} = u^{(k)}_{|\Gamma_1} & \text{on } \Gamma_1 \\ u_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1 \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\ u_2^{(k+1)} = u^{(k+1)}_{1|\Gamma_2} & \text{on } \Gamma_2 \\ u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2 \end{cases} . \tag{1.11}$$

The next approximate $u^{(k+1)}$ is then defined by

$$u^{(k+1)}(x) = \begin{cases} u_2^{(k+1)}(x) & \text{if } x \in \Omega_2 \\ u_1^{(k+1)}(x) & \text{if } x \in \Omega \setminus \Omega_2 \end{cases} .$$

It can be shown that the multiplicative Schwarz method (1.11) converges to a solution of problem (1.5), see [77, 78] and for a variational based proof consult [100]. In particular, there exist constants $c_1, c_2 \in (0, 1)$, which depend only on (Ω_1, Γ_2) and (Ω_2, Γ_1) respectively, such that for all $k \geq 0$

$$\|u_{|\Omega_1} - u_1^{(k+1)}\|_{L^\infty(\Omega_1)} \leq c_1^k c_2^k \|u - u^{(0)}\|_{L^\infty(\Gamma_1)}$$
$$\|u_{|\Omega_2} - u_2^{(k+1)}\|_{L^\infty(\Omega_2)} \leq c_1^{k+1} c_2^k \|u - u^{(0)}\|_{L^\infty(\Gamma_2)}.$$

Note that the constants c_1, c_2 depend on the size of the overlap and they can be quite close to one if the overlapping region is thin [78].

Variational formulation Set $(w, v) := \int_\Omega wv$, $a(w, v) := (\mathcal{L}w, v)$, and $H_0^1(\Omega_i) := \{v \in H_0^1(\Omega) : v = 0 \text{ in } \Omega \setminus \overline{\Omega_i}\}$ as closed subspaces of $H_0^1(\Omega)$ by extending their elements on Ω by 0. Moreover we define the energy

$$J(w, u) := \frac{1}{2}a(w, w) - (f, w) + a(u, w). \tag{1.12}$$

Let us rewrite (1.11) in the following form

$$\begin{cases} \mathcal{L}(u^{(k+1/2)} - u^{(k)}) = f - \mathcal{L}u^{(k)} & \text{in } \Omega_1 \\ u^{(k+1/2)} - u^{(k)} \in H_0^1(\Omega_1) \end{cases}$$

and

$$\begin{cases} \mathcal{L}(u^{(k+1)} - u^{(k+1/2)}) = f - \mathcal{L}u^{(k+1/2)} & \text{in } \Omega_2 \\ u^{(k+1)} - u^{(k+1/2)} \in H_0^1(\Omega_2). \end{cases}$$

The variational formulation of method (1.11) reads as follows: initialize $u^{(0)} \in H_0^1(\Omega)$ and for $k \geq 0$ solve

$$\begin{cases} w_1^{(k)} \in H_0^1(\Omega_1) : a(w_1^{(k)}, v_1) = (f, v_1) - a(u^{(k)}, v_1) \quad \text{for all } v_1 \in H_0^1(\Omega_1) \\ u^{(k+1/2)} = u^{(k)} + w_1^{(k)} \\ w_2^{(k)} \in H_0^1(\Omega_2) : a(w_2^{(k)}, v_2) = (f, v_2) - a(u^{(k+1/2)}, v_2) \quad \text{for all } v_2 \in H_0^1(\Omega_2) \\ u^{(k+1)} = u^{(k+1/2)} + w_2^{(k)} \end{cases} \tag{1.13}$$

or equivalently

$$\begin{cases} w_1^{(k)} = \arg\min_{w_1 \in H_0^1(\Omega_1)} J(w_1, u^{(k)}) \\ u^{(k+1/2)} = u^{(k)} + w_1^{(k)} \\ w_2^{(k)} = \arg\min_{w_2 \in H_0^1(\Omega_2)} J(w_2, u^{(k+1/2)}) \\ u^{(k+1)} = u^{(k+1/2)} + w_2^{(k)}. \end{cases} \tag{1.14}$$

From (1.13) we have

$$\begin{aligned} a(u^{(k+1/2)} - u^{(k)}, v_1) &= a(u - u^{(k)}, v_1), \quad u^{(k+1/2)} - u^{(k)} \in H_0^1(\Omega_1) \\ a(u^{(k+1)} - u^{(k+1/2)}, v_1) &= a(u - u^{(k+1/2)}, v_1), \quad u^{(k+1)} - u^{(k+1/2)} \in H_0^1(\Omega_2), \end{aligned}$$

which means

$$\begin{aligned} u^{(k+1/2)} - u^{(k)} &= P_1(u - u^{(k)}) \quad \text{for all } k \geq 0 \\ u^{(k+1)} - u^{(k+1/2)} &= P_2(u - u^{(k+1/2)}) \quad \text{for all } k \geq 0 \end{aligned}$$

or equivalently

$$\begin{aligned} u - u^{(k+1/2)} &= (I - P_1)(u - u^{(k)}) \quad \text{for all } k \geq 0 \\ u - u^{(k+1)} &= (I - P_2)(u - u^{(k+1/2)}) \quad \text{for all } k \geq 0, \end{aligned}$$

where $P_i : H_0^1(\Omega) \to H_0^1(\Omega_i)$ are orthogonal projections. From the latter immediately follows the error recursion formula

$$u - u^{(k+1)} = (I - P_2)(u - u^{(k+1/2)}) = (I - P_2)(I - P_1)(u - u^{(k)}) \quad \text{for all } k \geq 0. \tag{1.15}$$

Additive Schwarz Method

If we make the two steps (1.11) independent from each other, then we obtain the additive alternating Schwarz method, which computes the sequence of approximations by solving

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f & \text{in } \Omega_1 \\ u_1^{(k+1)} = u^{(k)}_{|\Gamma_1} & \text{on } \Gamma_1 \\ u_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1 \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{L}u_2^{(k+1)} = f & \text{in } \Omega_2 \\ u_2^{(k+1)} = u^{(k)}_{|\Gamma_2} & \text{on } \Gamma_2 \\ u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2 \end{cases} . \tag{1.16}$$

The next update $u^{(k+1)}$ is then defined by

$$u^{(k+1)}(x) = \begin{cases} u_1^{(k+1)}(x) & x \in \Omega \setminus \Omega_2 \\ u_1^{(k+1)}(x) + u_2^{(k+1)}(x) - u^{(k)}(x) & x \in \Omega_1 \cap \Omega_2 \\ u_2^{(k+1)}(x) & x \in \Omega \setminus \Omega_1 \end{cases} . \tag{1.17}$$

Variational Formulation The variational formulation of method (1.16) reads as

$$\begin{cases} w_1^{(k)} \in H_0^1(\Omega_1) : a(w_1^{(k)}, v_1) = (f, v_1) - a(u^{(k)}, v_1) & \text{for all } v_1 \in H_0^1(\Omega_1) \\ w_2^{(k)} \in H_0^1(\Omega_2) : a(w_2^{(k)}, v_2) = (f, v_2) - a(u^{(k)}, v_2) & \text{for all } v_2 \in H_0^1(\Omega_2) \\ u^{(k+1)} = u^{(k)} + w_1^{(k)} + w_2^{(k)} \end{cases} \tag{1.18}$$

or

$$\begin{cases} w_1^{(k)} = \arg\min_{w_1 \in H_0^1(\Omega_1)} J(w_1, u^{(k)}) \\ w_2^{(k)} = \arg\min_{w_2 \in H_0^1(\Omega_2)} J(w_2, u^{(k)}) \\ u^{(k+1)} = u^{(k)} + w_1^{(k)} + w_2^{(k)} \end{cases} , \tag{1.19}$$

where $J(w, u) = \frac{1}{2}a(w, w) - (f, w) + a(u, w)$. By relation (1.17) we verify that the original formulation (1.16) is equivalent to the variational formulation. Moreover from (1.18) we have that

$$\begin{aligned} a(w_1^{(k)}, v_1) &= a(u_1^{(k+1)} - u^{(k)}, v_1) = a(u - u^{(k)}, v_1) \\ a(w_2^{(k)}, v_2) &= a(u_2^{(k+1)} - u^{(k)}, v_2) = a(u - u^{(k)}, v_2) \end{aligned}$$

and hence we deduce

$$\begin{aligned} u_1^{(k+1)} - u^{(k)} &= P_1(u - u^{(k)}) \quad \text{for all } k \geq 0 \\ u_2^{(k+1)} - u^{(k)} &= P_2(u - u^{(k)}) \quad \text{for all } k \geq 0 \end{aligned}$$

or equivalently

$$\begin{aligned} u - u_1^{(k+1)} &= (I - P_1)(u - u^{(k)}) \quad \text{for all } k \geq 0 \\ u - u_2^{(k+1)} &= (I - P_2)(u - u^{(k)}) \quad \text{for all } k \geq 0. \end{aligned}$$

Then by using the update (1.17) we get the following error recursion formula:

$$u - u^{(k+1)} = u - u_1^{(k+1)} - u_2^{(k+1)} - u^{(k)} = (I - P_1 - P_2)(u - u^{(k)}) \quad \text{for all } k \geq 0. \quad (1.20)$$

Inspired by the variational formulation (1.14) and (1.19) of the multiplicative and additive Schwarz method, in [61] a minimization of a functional formed by a discrepancy term with respect to the data and by a ℓ_1-norm constraint by means of subspace correction is proposed. That is, the functional is minimized by alternately minimizing local problems that are restricted to suitable subspaces. We note that this problem is non-smooth, since a ℓ_1-term is present, but additive with respect to the proposed splitting and therefore can be included in the class of problems discussed in [21]. We recall this approach since it serves us as a model for subspace correction methods for non-differentiable problems.

1.3 Subspace Correction for ℓ_1-norm Minimization

The minimization of the ℓ_1-norm is well-understood to provide an effective way for reconstructing sparse signals from linear measurement [62]. It has been shown that the minimization of the ℓ_1-norm is very effective in several applications, such as compressed sensing [16, 17, 18, 52], image processing, and inverse problems [70, 110]. Let $\mathcal{H}$ be a real separable Hilbert space and for a countable index set Λ we define $\ell_p(\Lambda) := \{u = (u_\lambda)_{\lambda\in\Lambda} : (\sum_{\lambda\in\Lambda} |u_\lambda|^p)^{1/p}\}$ for $1 \leq p < \infty$. We are interested in the numerical minimization in $\ell_2(\Lambda)$ of the functional

$$J(u) := \|Tu - g\|_{\mathcal{H}}^2 + 2\alpha\|u\|_{\ell_1(\Lambda)}, \tag{1.21}$$

where $T : \ell_2(\Lambda) \to \mathcal{H}$ is a bounded linear operator, $g \in \mathcal{H}$ is a given observed datum, and $\alpha > 0$ is a fixed regularization parameter. In order to solve this minimization problem with respect to u one can take an *iterative thresholding algorithm* [42]: pick an initial $u^{(0)} \in \ell_2(\Lambda)$ ($u^{(0)} = 0$ is a good choice) and iterate

$$u^{(n+1)} = \mathbb{S}_\alpha(u^{(n)} + T^*(g - Tu^{(n)})), \quad n \geq 0, \tag{1.22}$$

where T^* denotes the adjoint operator of T and $\mathbb{S}_\alpha : \ell_2(\Lambda) \to \ell_2(\Lambda)$, defined componentwise by $\mathbb{S}_\alpha(v) = (S_\alpha v_\lambda)_{\lambda\in\Lambda}$ with

$$S_\alpha(v) = \begin{cases} v - \operatorname{sign}(v)\alpha & \text{if } |v| > \alpha \\ 0 & \text{otherwise} \end{cases},$$

is the so-called *soft-thresholding operator*. The strong convergence of the algorithm in (1.22) to minimizers of J is proven in [42]. In [13] it was shown that under additional conditions on the operator T or on minimizers of (1.21) the algorithm in

(1.22) converges linearly, although with a rather poor rate in general, see [62] for a broader discussion. There exist several alternative approaches, that promise to solve ℓ_1-minimization with fast convergence [44, 58, 74, 8]. One way to accelerate the speed of convergence of minimizing iterative soft-thresholding algorithms for large-scale problems was proposed in [61]. There a sequential and parallel domain decomposition method for ℓ_1-norm minimization was introduced and analyzed. We will explain now in more detail the main idea of this algorithm.

1.3.1 Sequential Algorithm

We decompose the index set Λ into two disjoint sets Λ_i, $i = 1, 2$, i.e., $\Lambda = \Lambda_1 \cup \Lambda_2$. Associated with this decomposition we define $V_i = \{u_\Lambda \in \ell_2(\Lambda) : \operatorname{supp}(u_\Lambda) \subset \Lambda_i\}$ for $i = 1, 2$. Then we minimize J in (1.21) by using the following alternating algorithm: pick an initial $V_1 \oplus V_2 \ni u_{\Lambda_1}^{(0)} + u_{\Lambda_2}^{(0)} := u^{(0)}$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} u_{\Lambda_1}^{(n+1)} \approx \arg\min_{u_{\Lambda_1}\in V_1} J(u_{\Lambda_1} + u_{\Lambda_2}^{(n)}) \\ u_{\Lambda_2}^{(n+1)} \approx \arg\min_{u_{\Lambda_2}\in V_2} J(u_{\Lambda_1}^{(n+1)}, u_{\Lambda_2}) \\ u^{(n+1)} := u_{\Lambda_1}^{(n+1)} + u_{\Lambda_2}^{(n+1)}, \end{cases} \tag{1.23}$$

where u_{Λ_i} is supported on Λ_i only, $i = 1, 2$. This algorithm is inspired by (1.14) and (1.19), but differently from the situations there, the energy (1.21) is now non-smooth. Nevertheless we observe that the ℓ_1-norm splits additively

$$\|u_{\Lambda_1} + u_{\Lambda_2}\|_{\ell_1(\Lambda)} = \|u_{\Lambda_1}\|_{\ell_1(\Lambda_1)} + \|u_{\Lambda_2}\|_{\ell_1(\Lambda_2)}$$

and hence the subproblems in (1.23) are of the same kind as the original problem (1.21), i.e., for example, for the problem on Λ_1 we have

$$\arg\min_{u_{\Lambda_1}\in V_1} J(u_{\Lambda_1} + u_{\Lambda_2}^{(n)}) = \arg\min_{u_{\Lambda_1}\in V_1} \|T_{\Lambda_1} u_{\Lambda_1} - (g - T_{\Lambda_2} u_{\Lambda_2}^{(n)})\|_{L^2(\Omega)}^2 + 2\alpha \|u_{\Lambda_1}\|_{\ell_1(\Lambda_1)},$$

where T_{Λ_i} is the restriction of the matrix T to the columns indexed by Λ_i. Moreover, this splitting results in a dimension reduction for each subproblem. For solving the subminimization problems of (1.23) we can use one of the before mentioned methods, for example, again the iterative thresholding algorithm:

$$u_{\Lambda_i}^{(n+1,\ell+1)} = \mathbb{S}_\alpha(u_{\Lambda_i}^{(n+1,\ell)} + T_{\Lambda_i}^*((g - T_{\Lambda_{\hat{i}}} u_{\Lambda_{\hat{i}}}^{(n)}) - T_{\Lambda_i} u_{\Lambda_i}^{(n+1,\ell)})), \quad \hat{i} \in \{1, 2\} \setminus \{i\}. \tag{1.24}$$

This leads to the following sequential algorithm: pick an initial $V_1 \oplus V_2 \ni u_{\Lambda_1}^{(0,L)} + u_{\Lambda_2}^{(0,M)} := u^{(0)}$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} \begin{cases} u_{\Lambda_1}^{(n+1,0)} = u_{\Lambda_1}^{(n,L)} \\ u_{\Lambda_1}^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(u_{\Lambda_1}^{(n+1,\ell)} + T_{\Lambda_1}^*((g - T_{\Lambda_2} u_{\Lambda_2}^{(n,M)}) - T_{\Lambda_1} u_{\Lambda_1}^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, L-1 \end{cases} \\ \begin{cases} u_{\Lambda_2}^{(n+1,0)} = u_{\Lambda_2}^{(n,M)} \\ u_{\Lambda_2}^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(u_{\Lambda_2}^{(n+1,\ell)} + T_{\Lambda_2}^*((g - T_{\Lambda_1} u_{\Lambda_1}^{(n+1,L)}) - T_{\Lambda_2} u_{\Lambda_2}^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, M-1 \end{cases} \\ u^{(n+1)} := u_{\Lambda_1}^{(n+1,L)} + u_{\Lambda_2}^{(n+1,M)}. \end{cases} \tag{1.25}$$

Note, that we perform only a finite number L and M of inner iterations. However, for any choice of L and M this algorithm produces a sequence $(u^{(n)})_n$ such that $J(u^{(n)})$ is monotonically decreasing. Moreover, its convergence to a strong minimizer of the functional (1.21) is proven [61]. Nothing is known about the rate of convergence, which is still an open problem. However, for instance, Vonesch and Unser [116] used the algorithm in (1.23) in order to provide some preconditioning effects when dealing with deblurring problems.

Great advantages of this subspace correction algorithm are that we can solve instead of one large problem several smaller problems, which might lead to an acceleration of convergence due to preconditioning effects with a reduction of overall computational cost, and that it can be easily parallelized.

1.3.2 Parallel Algorithm

Algorithm (1.25) can be modified to a parallel version by just substituting $u_{\Lambda_1}^{(n+1,L)}$ by $u_{\Lambda_1}^{(n,L)}$ in the second inner iterations, which makes the subminimization problems on Λ_1 and Λ_2 independent from each other. Additionally, in order to guarantee the monotonicity of the sequence $J(u^{(n)})$, we need to replace the update $u^{(n+1)} := u_{\Lambda_1}^{(n+1,L)} + u_{\Lambda_2}^{(n+1,M)}$ by $u^{(n+1)} := \frac{u_{\Lambda_1}^{(n+1,L)} + u_{\Lambda_2}^{(n+1,M)} + u^{(n)}}{2}$, which is the average between the current and the previous iteration. Then we obtain the following parallel algorithm: pick an initial $V_1 \oplus V_2 \ni u_{\Lambda_1}^{(0,L)} + u_{\Lambda_2}^{(0,M)} := u^{(0)}$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} \begin{cases} u_{\Lambda_1}^{(n+1,0)} = u_{\Lambda_1}^{(n,L)} \\ u_{\Lambda_1}^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(u_{\Lambda_1}^{(n+1,\ell)} + T_{\Lambda_1}^*((g - T_{\Lambda_2} u_{\Lambda_2}^{(n,M)}) - T_{\Lambda_1} u_{\Lambda_1}^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, L-1 \end{cases} \\ \begin{cases} u_{\Lambda_2}^{(n+1,0)} = u_{\Lambda_2}^{(n,M)} \\ u_{\Lambda_2}^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(u_{\Lambda_2}^{(n+1,\ell)} + T_{\Lambda_2}^*((g - T_{\Lambda_1} u_{\Lambda_1}^{(n,L)}) - T_{\Lambda_2} u_{\Lambda_2}^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, M-1 \end{cases} \\ u^{(n+1)} := \frac{u_{\Lambda_1}^{(n+1,L)} + u_{\Lambda_2}^{(n+1,M)} + u^{(n)}}{2}. \end{cases}$$

The convergence to a strong minimizer of the functional J of the parallel algorithm is also proven in [61].

1.3.3 Multidomain Splitting

The above described subspace correction algorithm is not restricted to a decomposition into 2 subspaces, but can be generalized to an algorithm for multiple decompositions. We split now the index set Λ into multiple disjoint sets Λ_i, $i = 1, 2, \dots, \mathcal{N}$, such that $\Lambda = \bigcup_{i=1}^{\mathcal{N}} \Lambda_i$. Associated with this decomposition we define $V_i = \{u_\Lambda \in \ell_2(\Lambda) : \operatorname{supp}(u_\Lambda) \subset \Lambda_i\}$ for $i = 1, 2, \dots, \mathcal{N}$. Then we minimize J by using the following alternating algorithm: pick an initial $V_1 \oplus \ldots \oplus V_{\mathcal{N}} \ni u_{\Lambda_1}^{(0,L)} + \ldots + u_{\Lambda_{\mathcal{N}}}^{(0,L_{\mathcal{N}})} := u^{(0)}$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} \begin{cases} u_{\Lambda_1}^{(n+1,0)} = u_{\Lambda_1}^{(n,L_1)} \\ u_{\Lambda_1}^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(u_{\Lambda_1}^{(n+1,\ell)} + T_{\Lambda_1}^* ((g - \sum_{i=2}^{\mathcal{N}} T_{\Lambda_i} u_{\Lambda_i}^{(n,L_i)}) - T_{\Lambda_1} u_{\Lambda_1}^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, L_1 - 1 \end{cases} \\ \dots \\ \begin{cases} u_{\Lambda_{\mathcal{N}}}^{(n+1,0)} = u_{\Lambda_{\mathcal{N}}}^{(n,L_{\mathcal{N}})} \\ u_{\Lambda_{\mathcal{N}}}^{(n+1,\ell+1)} = \mathbb{S}_\alpha \left(u_{\Lambda_{\mathcal{N}}}^{(n+1,\ell)} + T_{\Lambda_{\mathcal{N}}}^* ((g - \sum_{i=1}^{\mathcal{N}-1} T_{\Lambda_i} u_{\Lambda_i}^{(n+1,L_i)}) - T_{\Lambda_{\mathcal{N}}} u_{\Lambda_{\mathcal{N}}}^{(n+1,\ell)}) \right) \\ \ell = 0, \dots, L_{\mathcal{N}} - 1 \end{cases} \\ u^{(n+1)} := \frac{\sum_{i=1}^{\mathcal{N}} u_{\Lambda_i}^{(n+1,L_i)} + (\mathcal{N}-1) u^{(n)}}{\mathcal{N}}. \end{cases}$$

The monotonicity of the energy with respect to the iterations and the convergence to an expected minimizer is ensured [61].

1.4 On this Work

Motivated by the above mentioned subspace correction method for ℓ_1-norm minimization we are interested to extend such a splitting strategy for *non-smooth* and *non-additive* problems. In particular, in this thesis we are concerned with introducing convergent sequential and parallel subspace correction and domain decomposition methods for solving functionals of the type (1.4), i.e., formed by an L^2-discrepancy term and a total variation constraint.

Note that in comparison to the ℓ_1-norm, the total variation is both non-smooth and non-additive with respect to a non-overlapping domain decomposition, since the total variation of a function on the whole domain equals the sum of the total variation on the subdomains plus the size of the possible jumps at the interfaces, see formula (2.6). Thus one encounters additional difficulties in showing convergence of such decomposition strategies to global minimizers. In particular, we stress very clearly that well-known approaches as in [21, 30, 108, 109] are not directly applicable to this problem, because they either address additive problems or smooth convex minimizations, which is *not* the case of total variation minimization. We emphasize that the successful convergence of such alternating algorithms is far from being obvious for non-smooth and non-additive problems, as many counterexamples can be constructed. Moreover, for total variation minimization, the interesting solutions may be discontinuous, e.g., along curves in 2D. These discontinuities may cross the interfaces of the domain decomposition patches. Hence, the crucial difficulty is the correct numerical treatment of interfaces, with the preservation of crossing discontinuities and the correct matching where the solution is continuous instead.

Contribution of this thesis

This thesis is putting together the contributions of the papers [63, 64, 75, 76] which we distribute into the following two chapters:

Chapter 3 The work [66] was generally addressed to abstract subspace correction methods for the minimization of functionals formed by a discrepancy term and a non-smooth and non-additive constraint. An implementation of this algorithm is suggested that is guaranteed to converge and to decrease the objective energy $\mathcal{J}$ monotonically. The convergence to minimizers of $\mathcal{J}$ could be only proven under technical conditions on the interface of the subdomains, which are in general not fulfilled. In fact we will state a simple counterexample in 2-dimensions taken from [118], where we show that for a non-smooth and non-additive energy a subspace correction method does not necessarily converge to the right minimal solution. Additionally and as the first original result we show a more sophisticated counterexample with respect to the total variation minimization as described below.

Inspired by the work of Vonesch and Unser [116], we adapt and specify the subspace correction algorithm from [66] to the case of an orthogonal wavelet space decomposition and for deblurring problems. We show additional properties of the limit of the sequence produced by the subspace correction algorithm for L^2/TV-minimization problems and obtain an additional condition under which the obtained limit is indeed the expected minimizer. Nevertheless, this condition cannot be ensured to hold always. In particular we are able to construct a counterexample, which shows that in general we cannot expect convergence of the algorithm to a minimizer of the objective functional. Despite this quite special negative result, we show in this chapter that an orthogonal wavelet space decomposition for deblurring problems works in practice very efficiently. With the help of the newly obtained condition of convergence, we are indeed able to show in our numerical examples that the sequence produced by this algorithm in fact numerically converges to the expected minimizer.

Chapter 4 As the second relevant result of the thesis, despite the partially negative results provided by our counterexamples of Chapter 3, we prove that appropriate domain decomposition algorithms for total variation minimization do converge to minimizers of $\mathcal{J}$. We introduce overlapping domain decomposition algorithms for total variation minimization that eventually provides us with a framework in which we are able to prove successfully its convergence to minimizers of $\mathcal{J}$, both in its sequential and parallel form. Further the subspace correction algorithm of [66] (also presented in the Chapter 3), can be particularly modified to work for a non-overlapping domain decomposition for the minimization of the total variation. In a very general framework it was not possible to show that the algorithm is indeed converging to the expected minimizer. However the numerical results seemed very promising and converging to the right solution. We close this gap between the theoretical analysis and the numerical experiments by proving that in a discrete setting the proposed algorithm always converges to a minimizer of the discretized functional. Finally let us stress that these are the first methods that address a domain decomposition strategy for total variation minimization with a formal theoretical justification of convergence.

In both the non-overlapping and the overlapping algorithm we have to solve constrained minimization problems on each subdomain. We suggest to minimize these subproblems either by the so-called iterative oblique thresholding algorithm from [66] or by the Bregmanized Operator Splitting - Split Bregman algorithm, which combines the recently introduced Bregmanized Operator Splitting and the Split Bregman method. We compare both approaches with respect to computational time and efficiency for the example of the non-overlapping domain decomposition method for total variation minimization.

Chapter 2

Preliminaries

In this chapter we introduce basic concepts of convex analysis and functions of bounded variations, which will be useful to us in the sequel.

2.1 Convex Functions

Definition 2.1.1. *Let V be a vector space over $\mathbb{R}$.*

(a) The domain of a function $F : V \to \bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is defined as the set

$$\mathrm{Dom}(F) = \{v \in V : F(v) \neq \infty\}.$$

(b) A set $U \subset V$ is said to be convex *if, for all $u, v \in U$ and all $\lambda \in [0, 1]$ we have that*

$$\lambda u + (1 - \lambda)v \in U.$$

(c) Additionally let $U \subset V$ a convex space and $F : U \to \bar{\mathbb{R}}$. Then F is said to be convex *if for every $u, v \in U$*

$$F(\lambda u + (1 - \lambda)v) \leq \lambda F(u) + (1 - \lambda)F(v)$$

for all $\lambda \in [0, 1]$, whenever the right-hand side is defined.

(d) A function $F : V \to \mathbb{R}$ is said to be sublinear *if for all $u, v \in V$*

$$F(u + v) \leq F(u) + F(v).$$

(e) We say that a function $F : V \to \mathbb{R}$ is 1-homogeneous *if for all $u \in V$ and $\lambda \in \mathbb{R}$*

$$F(\lambda u) = |\lambda| F(u).$$

A convex function $F : V \to \bar{\mathbb{R}}$ is called *proper*, if $\mathrm{Dom}(F) \neq \emptyset$ and $F(u) > -\infty$ for all $u \in V$. We recall that a *topological vector space* is a vector space V endowed with a topology such that the maps $(u, v) \mapsto u + v$ of $V \times V$ into V and $(\lambda, u) \mapsto \lambda u$ of $\mathbb{R} \times V$ into V are continuous. Moreover a topological vector space is called *locally convex* if the origin possesses a fundamental system of convex neighborhoods.

Definition 2.1.2. *Let V be a locally convex space. A function $F : V \to \bar{\mathbb{R}}$ is called* lower semicontinuous (l.s.c.) *on V if for every convergent sequence $u^{(n)} \to \hat{u}$ we have*

$$\liminf_{u^{(n)} \to \hat{u}} F(u^{(n)}) \geq F(\hat{u}).$$

2.2 Duality in Convex Optimization

2.2.1 Conjugate Functions

In this section we recall duality results in convex analysis, see for example [53, 101]. Henceforth we let V and V^* be two vector spaces placed in the duality by a bilinear pairing denoted by $\langle \cdot, \cdot \rangle$. We would like to start by recalling a useful result.

Proposition 2.2.1. *Let $F : V \to \bar{\mathbb{R}}$, then the following properties are equivalent:*

(i) F is the pointwise supremum of a family of continuous affine functions;

(ii) F is a convex l.s.c. function from V into $\bar{\mathbb{R}}$, and if F takes the value $-\infty$ then F is identically equal to $-\infty$.

Let us introduce the definition of the *conjugate (or Legendre transform)* of a function.

Definition 2.2.2. *Let $F : V \to \bar{\mathbb{R}}$ be a function. The* conjugate function (or Legendre transform) *$F^* : V^* \to \mathbb{R}$ is defined by*

$$F^*(u^*) = \sup_{u \in V} \{\langle u, u^* \rangle - F(u)\}. \tag{2.1}$$

From this definition we see that F^* is the pointwise supremum of continuous affine functions and thus, according to Proposition 2.2.1, convex and lower semicontinous. The conjugate function F^* is additionally proper if and only if F is proper. We can repeat the process in (2.1), which leads to the *biconjugate* of a function.

Definition 2.2.3. *Let F^* be the conjugate function of $F : V \to \bar{\mathbb{R}}$. Then the function $F^{**} : V \to \bar{\mathbb{R}}$ defined by*

$$F^{**}(u) = \sup_{u^* \in V^*} \{\langle u, u^* \rangle - F^*(u^*)\}$$

is called the biconjugate *of F.*

We have the following properties

Proposition 2.2.4. *Let $F : V \to \bar{\mathbb{R}}$, then we observe the following properties:*

(i) $F^{**} \leq F$ *and if* F *is convex, proper, and l.s.c., then* $F^{**} = F$*,*

(ii) $F^{***} = F^*$*.*

2.2.2 Computation of Conjugate Functions

Let us denote by $\| \cdot \|$ the norm of V and by $\| \cdot \|_*$ the norm of V^*. Now we will give two examples of computing the conjugate function of given convex functions.

Example 1: Let us compute the conjugate function of the convex function $F_1 : V \to \mathbb{R}$ that is defined by $F_1(u) = \|u - g\|^2$, where $g \in V$ is fixed. From Definition 2.2.2 we have

$$F_1^*(u^*) = \sup_{u \in V}\{\langle u, u^* \rangle - F_1(u)\} = \sup_{u \in V}\{\langle u, u^* \rangle - \langle u - g, u - g \rangle\}.$$

We set $G(u) := \langle u, u^* \rangle - \langle u - g, u - g \rangle$. To get the maximum of G we compute the Gâteaux-differential (see Definition 2.3.1) at u of G,

$$G'(w_0) = u^* - 2(u - g) = 0$$

and we set it to zero $G'(u) = 0$, since $G''(u) < 0$, and we get $u = \frac{u^*}{2} + g$, being the maximizer of G. Thus we have that

$$F_1^*(u^*) = \sup_{u \in V} G(u) = \left\langle \frac{u^*}{4} + g, u^* \right\rangle.$$

Example 2: Now we are going to compute the conjugate function of $F_2 : V \to \mathbb{R}$ that is defined by $F_2(u) = 2\alpha\varphi(\|u\|)$, where $\varphi : \mathbb{R} \to \mathbb{R}$ is a convex function and $\alpha > 0$. From Definition 2.2.2 we have

$$\begin{aligned} F_2^*(u^*) &= \sup_{u \in V}\{\langle u, u^* \rangle - 2\alpha\varphi(\|u\|)\} \\ &= \sup_{t \geq 0} \sup_{\substack{u \in V \\ \|u\| = t}} \{\langle u, u^* \rangle - 2\alpha\varphi(\|u\|)\} \\ &= \sup_{t \geq 0}\{t\|u^*\|_* - 2\alpha\varphi(t)\}. \end{aligned}$$

If φ were an even function, i.e., $\varphi(t) = \varphi(-t)$ for all $t \in \mathrm{Dom}(\varphi)$, then

$$\begin{aligned} \sup_{t \geq 0}\{t\|u^*\|_* - 2\alpha\varphi(t)\} &= \sup_{t \in \mathbb{R}}\{t\|u^*\|_* - 2\alpha\varphi(t)\} \\ &= 2\alpha \sup_{t \in \mathbb{R}} \left\{ \frac{t\|u^*\|_*}{2\alpha} - \varphi(t) \right\} \\ &= 2\alpha\varphi^* \left(\frac{\|u^*\|_*}{2\alpha} \right), \end{aligned}$$

where φ^* is the conjugate function of φ.

Unfortunately φ is not even in general. To overcome this difficulty we have to choose a function which is equal to $\varphi(s)$ for $s \geq 0$ and does not change the supremum for $s < 0$. For instance, one can choose $\varphi_1(s) = \varphi(|s|)$ for $s \in \mathbb{R}$. Then we have

$$\begin{aligned}\sup_{t\geq 0}\{t\|u^*\|_* - 2\alpha\varphi(t)\} &= \sup_{t\in\mathbb{R}}\{t\|u^*\|_* - 2\alpha\varphi_1(t)\}\\ &= 2\alpha \sup_{t\in\mathbb{R}}\left\{t\frac{\|u^*\|_*}{2\alpha} - \varphi_1(t)\right\}\\ &= 2\alpha\varphi_1^*\left(\frac{\|u^*\|_*}{2\alpha}\right),\end{aligned}$$

where φ_1^* is the conjugate function of φ_1. Note that one can also choose $\varphi_1(s) = \varphi(s)$ for $s \geq 0$ and $\varphi_1(s) = +\infty$ for $s < 0$.

2.2.3 The Dual Problem

Let $F : V \to \bar{\mathbb{R}}$ be a function. We are now interested in the minimization of the problem

$$\inf_{u\in V} F(u). \tag{$\mathbf{P}$}$$

We call problem $(\mathbf{P})$ the *primal problem* and it is said to be *non-trivial* if there exists $u_0 \in V$ such that

$$F(u_0) < +\infty.$$

The infimum in $(\mathbf{P})$ is denoted by $\inf(\mathbf{P})$. Let us introduce two other topological vector spaces Y and Y^* placed in the duality by a bilinear pairing $\langle\cdot,\cdot\rangle$. We will denote the pairing between V and V^* as well as the pairing between Y and Y^* by $\langle\cdot,\cdot\rangle$, since there is in general no possibility of ambiguity. Now consider a function $\phi : V \times Y \to \bar{\mathbb{R}}$ such that

$$\phi(u, 0) = F(u)$$

and for every $p \in Y$ we consider the minimization problem

$$\inf_{u\in V} \phi(u, p).$$

Obviously for $p = 0$ the latter problem is the same as $(\mathbf{P})$.

We define now the *dual problem* of $(\mathbf{P})$: let $\phi^* : V^* \times Y^* \to \bar{\mathbb{R}}$ be the conjugate function of ϕ in the duality between $V \times Y$ and $V^* \times Y^*$, i.e., $\phi^*(u^*, p^*) = \sup_{u\in V, p\in Y}\{\langle u^*, u\rangle + \langle p^*, p\rangle - \phi(u, p)\}$. Then we call

$$\sup\{-\phi^*(0, p^*)\} \tag{$\mathbf{P}^*$}$$

the dual problem of $(\mathbf{P})$ with respect to ϕ (or with respect to the given perturbation p). We denote the supremum of $(\mathbf{P}^*)$ by $\sup(\mathbf{P}^*)$.

An Important Special Case

Assume $T : V \to Y$ is a continuous linear operator with adjoint $T^* : Y^* \to V^*$ and the function to be minimized can be written as

$$F(u) = G(u, Tu),$$

where G is a function of $V \times Y$ into $\bar{\mathbb{R}}$. Then the primal problem $(\mathbf{P})$ takes the form

$$\inf_{u \in V} G(u, Tu). \tag{2.2}$$

In this case we write the function ϕ as

$$\phi(u, p) = G(u, Tu - p).$$

Now let us determine the dual problem of (2.2): we have

$$\begin{aligned}\phi^*(0, p^*) &= \sup_{u \in V, p \in Y} \{\langle p^*, p\rangle - G(u, Tu - p)\}\\ &= \sup_{u \in V} \sup_{p \in Y} \{\langle p^*, p\rangle - G(u, Tu - p)\}.\end{aligned}$$

For a fixed $u \in V$, we substitute $q = Tu - p$ and obtain

$$\begin{aligned}\phi^*(0, p^*) &= \sup_{u \in V} \sup_{q \in Y} \{\langle p^*, Tu\rangle - \langle p^*, q\rangle - G(u, q)\}\\ &= \sup_{u \in V, q \in Y} \{\langle T^*p^*, u\rangle - \langle p^*, q\rangle - G(u, q)\} = G^*(T^*p^*, -p^*),\end{aligned}$$

where G^* denoted the conjugate function of G. Thus the dual problem reads as

$$\sup_{p^* \in Y^*} \{-G^*(T^*p^*, -p^*)\}.$$

Theorem 2.2.5. *Let us as assume that G is convex, that $\inf(\mathbf{P})$ is finite, and that there exists $u_0 \in V$ such that $G(u_0, Tu_0) < +\infty$ and the function $p \mapsto G(u_0, p)$ is continuous at Tu_0. Then problem (2.2) is stable:*

$$\inf(\mathbf{P}) = \sup(\mathbf{P}^*),$$

and $(\mathbf{P}^*)$ has at least one solution $\bar{p}^*$.

If the function G can be additionally decomposed into the form

$$G(u, Tu) = G_1(u) + G_2(Tu),$$

with $G_1 : V \to \bar{\mathbb{R}}$ and $G_2 : Y \to \bar{\mathbb{R}}$, then the problem $(\mathbf{P})$ can be written as

$$\inf_{u \in V} \{G_1(u) + G_2(Tu)\}.$$

It is easy to verify that the dual problem is given by

$$\sup_{p^* \in Y^*} \{-G_1^*(T^*p^*) - G_2^*(-p^*)\},$$

where $G_1^* : V^* \to \bar{\mathbb{R}}$ and $G_2 : Y^* \to \bar{\mathbb{R}}$ are the conjugate functions of G_1 and G_2 respectively.

2.3 Subdifferentiability

In this section we extracted results from [53].

Definition 2.3.1. *Let V be a locally convex space and let F be a function of V into $\bar{\mathbb{R}}$. The* directional derivative *of F at $u \in V$ in the direction $v \in V$ is defined as the limit, if it exists,*

$$F'(u;v) = \lim_{\lambda \to 0_+} \frac{F(u+\lambda v) - F(u)}{\lambda}.$$

We say that F is Gâteaux-differentiable *at $u \in V$, and is denoted $F'(u) \in V^*$, if the above limit exists for every $v \in V$ and*

$$F'(u,v) = \langle v, F'(u) \rangle.$$

There exists functions in which the above limit does not exist, which means that these functions are not differentiable. For such functions we introduce a more general concept of differentiability, called *subdifferentiability.*

Definition 2.3.2. *Let V be a locally convex space, V^* its topological dual, $\langle \cdot, \cdot \rangle$ the bilinear canonical pairing over $V \times V^*$, and $F : V \to \bar{\mathbb{R}}$. The* subdifferential *of F at $u \in V$ is defined as the set valued function*

$$\partial F(u) := \begin{cases} \emptyset & \text{if } F(u) = \infty \\ \{u^* \in V^* : \langle u^*, v - u \rangle + F(u) \leq F(v) \quad \text{for all } v \in V\} & \text{otherwise} \end{cases}.$$

It is obvious from this definition that $0 \in \partial F(u)$ if and only if u is a minimizer of F. Since we deal with several spaces it will turn out to be useful to distinguish sometimes in which space the subdifferential is defined by imposing a subscript $\partial_V F$ for the subdifferential considered on the space V.

Proposition 2.3.3. *For every function $F : V \to \bar{\mathbb{R}}$ we have*

$$u^* \in \partial F(u) \Rightarrow u \in \partial F^*(u^*).$$

If, further, F is convex, l.s.c., and proper, we have

$$u^* \in \partial F(u) \Longleftrightarrow v \in \partial F^*(u^*). \tag{2.3}$$

2.3.1 Subdifferential Calculus

Let V be a locally convex space, $F : V \to \bar{\mathbb{R}}$, and $\lambda > 0$. At every point $u \in V$, we have

$$\partial(\lambda F)(u) = \lambda \partial F(u).$$

Moreover, let $F_1, F_2 : V \to \mathbb{R}$. At every point $u \in V$, we have

$$\partial(F_1 + F_2)(u) \supset \partial F_1(u) + \partial F_2(u).$$

Having an equality in the latter relation is far from being always fulfilled. However, there is a simple case where it holds:

Proposition 2.3.4. *Let F_1 and F_2 be convex, l.s.c., and proper. If there exists a point $\bar{u} \in \operatorname{Dom} F_1 \cap \operatorname{Dom} F_2$ where F_1 is continuous, then we have for all $u \in V$*

$$\partial(F_1 + F_2)(u) = \partial F_1(u) + \partial F_2(u).$$

Let us consider now the subdifferential of a composite function.

Proposition 2.3.5. *Let V and Y be two locally convex sets with topological duals V^* and Y^*, $T : V \to Y$ with adjoint T^*, $F : Y \to \mathbb{R}$ a convex, l.s.c., and proper function, and $F \circ T : V \to \mathbb{R}$ also a convex, l.s.c., and proper function. If there is a point $T\bar{u}$, for $\bar{u} \in V$, where F is continuous and finite, then for all points $u \in V$, we have*

$$\partial(F \circ T)(u) = T^* \partial F(Tu).$$

2.4 Basic Definitions on Measures

The following concepts and results are from [2, 60].

Definition 2.4.1. *Let X be a nonempty set and let $\mathcal{E}$ be a collection of subsets of X.*

(a) We say that $\mathcal{E}$ is an algebra *if $\emptyset \in \mathcal{E}$, $E_1 \cup E_2 \in \mathcal{E}$, and $X \setminus E_1 \in \mathcal{E}$ whenever $E_1, E_2 \in \mathcal{E}$.*

(b) We say that an algebra $\mathcal{E}$ is a σ-algebra *if for any sequence $(E_h) \subset \mathcal{E}$ its union $\bigcup_h E_h$ belongs to $\mathcal{E}$.*

(c) For any collection $\mathcal{G}$ of subsets of X, the σ-algebra generated by $\mathcal{G}$ is the smallest σ-algebra containing $\mathcal{G}$. If (X, τ) is a topological space, we denote by $\mathcal{B}(X)$ the σ-algebra of Borel subsets of X, i.e., the σ-algebra generated by the open subsets of X.

(d) If $\mathcal{E}$ is a σ-algebra in X, then we call the pair $(X, \mathcal{E})$ a measure space*.*

Let us define a positive measure.

Definition 2.4.2. *Let $(X, \mathcal{E})$ be a measure space and $\mu : \mathcal{E} \to [0, +\infty]$. We say that μ is a* positive measure *if $\mu(\emptyset) = 0$ and μ is σ-additive on $\mathcal{E}$, i.e., for any sequences (E_h) of pairwise disjoint elements of $\mathcal{E}$*

$$\mu\left(\bigcup_{h=0}^{\infty} E_h\right) = \sum_{h=0}^{\infty} \mu(E_h).$$

We say that μ is finite if $\mu(X) < +\infty$.

We now give the definition of real and vector measures.

Definition 2.4.3. *Let $(X, \mathcal{E})$ be a measure space and let $m \in \mathbb{N}$, $m \geq 1$. We say that $\mu : \mathcal{E} \to \mathbb{R}^m$ is a* measure *if $\mu(\emptyset) = 0$ and for any sequence (E_h) of pairwise disjoint elements of $\mathcal{E}$*

$$\mu\left(\bigcup_{h=0}^{\infty} E_h\right) = \sum_{h=0}^{\infty}(E_h).$$

If $m = 1$ we say that μ is a real measure*, if $m > 1$ we say that μ is a* vector measure*.*

A topological space X is called *compact* if every open cover of X has a finite subcover, i.e., for every collection $\{U_\alpha\}_{\alpha \in \Lambda}$ of open subsets of X such that $X = \bigcup_{\alpha \in \Lambda} U_\alpha$ there is a finite subset $\tilde{\Lambda}$ of Λ such that $X = \bigcup_{\alpha \in \tilde{\Lambda}} U_\alpha$. Moreover, if every point of X has a compact neighborhood, then we call X *locally compact*. We say X is *sequentially compact* if every sequence in X has a convergent subsequence. A topological space X is called *separable* if it has a countable dense subset, i.e., there is a subset $E \subset X$ such that $\operatorname{card}(E) \leq \operatorname{card}(\mathbb{N})$ and $\operatorname{cl} E = X$, where $\operatorname{card}(E)$ denotes the cardinality of the set E and $\operatorname{cl} E$ is the closure of E.

Definition 2.4.4. *Let X be a locally compact and separable metric space, $\mathcal{B}(X)$ its Borel σ-algebra, and consider the measure space $(X, \mathcal{B}(X))$. A (real or vector) set function defined on the relatively compact Borel subsets of X that is a measure on $(K, \mathcal{B}(K))$ for every compact set $K \subset X$ is called a* (real or vector) Radon measure *on X.*

Definition 2.4.5. *Let $k \in [0, +\infty]$, Λ any index set, and $E \subset \mathbb{R}^N$ for $N \in \mathbb{N}$. The* k-dimensional Hausdorff measure *of E is given by*

$$\mathcal{H}_k(E) := \lim_{\delta \to 0} \mathcal{H}_{k,\delta}(E),$$

where for $0 < \delta \leq +\infty$, $\mathcal{H}_{k,\delta}(E)$ is defined by

$$\mathcal{H}_{k,\delta}(E) := \frac{\omega_k}{2^k} \inf\left\{\sum_{i \in \Lambda} |\operatorname{diam}(E_i)|^k, \operatorname{diam}(E_i) < \delta, E = \bigcup_{i \in \Lambda} E_i\right\}$$

for finite or countable covers $(E_i)_{i \in \Lambda}$, where we denoted the diameter of the set E_i by $\operatorname{diam}(E_i)$*. The normalization factor $\omega_k = \pi^{k/2}\Gamma(1+k/2)$, where $\Gamma(t) = \int_0^\infty s^{t-1}e^{-s}ds$ is the Gamma function.*
The Hausdorff dimension of E is defined by

$$\mathcal{H} - \dim(E) := \inf\{k \geq 0 : \mathcal{H}_k(E) = 0\}.$$

2.5 Distributional Derivative

In the sequel let $\Omega \subset \mathbb{R}^N$, $N \geq 1$, be an open set. We define by $C_c^\infty(\Omega)$ the set of all C^∞-functions whose support is compact and contained in Ω. A distribution on Ω is a continuous linear functional on $C_c^\infty(\Omega)$. We denote the space of all distributions on Ω by $\mathcal{D}'(\Omega)$.

Definition 2.5.1. *For $F \in \mathcal{D}'(\Omega)$ and for any multi-index β we define the* distributional derivative $\partial^\beta F \in \mathcal{D}'(\Omega)$ *by*

$$\langle \partial^\beta F, \phi \rangle_{\mathcal{D}' \times C_c^\infty} = (-1)^{|\beta|} \langle F, \partial^\beta \phi \rangle_{\mathcal{D}' \times C_c^\infty} \quad \text{for all } \phi \in C_c^\infty(\Omega),$$

where $\langle \cdot, \cdot \rangle_{\mathcal{D}' \times C_c^\infty}$ denotes the pairing between $\mathcal{D}'(\Omega)$ and $C_c^\infty(\Omega)$.

For more details we refer to [60].

2.6 Functions of Bounded Variation

In image processing one is interested to recover and to preserve discontinuities in images. Using classical Sobolev spaces, as $W^{1,1}$ (the Sobolev space of L^1-functions with L^1-distributional derivatives), does not allow to take such phenomena into account, since the gradient of a Sobolev function is again a function. If u is discontinuous, then its gradient has to be understood as a *measure*. Therefore we introduce the space of bounded variation functions, which is adapted to this situation.

Definition 2.6.1. *Let $u \in L^1(\Omega)$, then u is a* function of bounded variation in Ω *if the distributional derivative of u is representable by a finite Radon measure in Ω, i.e., if*

$$\int_\Omega u \frac{\partial \phi}{\partial x_i} dx = - \int_\Omega \phi d D_i u \quad \text{for all } \phi \in C_c^\infty(\Omega),\ i = 1, \dots, N$$

for some $\mathbb{R}^N$-valued measure $Du = (D_1 u, \dots, D_N u)$ in Ω. The vector space of all functions of bounded variation in Ω is denoted by $BV(\Omega)$.

The variation of a function is defined as follows:

Definition 2.6.2. *For $u \in L^1_{loc}(\Omega)$, the* variation of u in Ω *is defined by*

$$V(u, \Omega) := \sup \left\{ \int_\Omega u \operatorname{div} \varphi \, dx : \varphi \in \left[C_c^1(\Omega) \right]^N, \|\varphi\|_\infty \leq 1 \right\}.$$

A simple integration by parts proves that $V(u, \Omega) = \int_\Omega |\nabla u| dx$ if $u \in C^1(\Omega)$ and in fact also if $u \in W^{1,1}(\Omega)$. Let us recall the definition of the total variation of a measure:

Definition 2.6.3. *Let X be a nonempty set and $\mathcal{E}$ be a σ-algebra on X. If μ is a measure, then its total variation $|\mu|$ for every $E \in \mathcal{E}$ is defined as*

$$|\mu|(E) := \sup \left\{ \sum_{h=0}^\infty |\mu(E_h)| : E_h \in \mathcal{E} \text{ pairwise disjoint, } E = \bigcup_{h=0}^\infty E_h \right\}.$$

The space of bounded variation functions can be characterized by the total variation $|Du|(\Omega)$. By this definition we can characterize the space of bounded variation functions by the total variation $|Du|(\Omega)$ in the following way:

Proposition 2.6.4. *Let $u \in L^1_{loc}(\Omega)$. Then, u belongs to $BV(\Omega)$ if and only if $V(u,\Omega) < \infty$. In addition, $V(u,\Omega)$ coincides with $|Du|(\Omega)$ for any $u \in BV(\Omega)$ and $u \mapsto |Du|(\Omega)$ is l.s.c. in $BV(\Omega)$ with respect to the $L^1_{loc}(\Omega)$ topology.*

The space $BV(\Omega)$ endowed with the norm

$$\|u\|_{BV} := \int_\Omega |u| dx + |Du|(\Omega) \tag{2.4}$$

is a Banach space. Moreover we have the following useful result.

Theorem 2.6.5. *Let $u \in L^1(\Omega)$. Then $u \in BV(\Omega)$ if and only if there exists a sequence $(v_n)_n \subset C^\infty(\Omega)$ converging to u in $L^1(\Omega)$ and satisfying*

$$C := \lim_{n\to+\infty} \int_\Omega |\nabla v_n| dx < \infty. \tag{2.5}$$

Moreover, the least constant C in (2.5) *is $|Du|(\Omega)$.*

Definition 2.6.6. *Let $u, u_h \in BV(\Omega)$. We say that $(u_n)_n$* weakly-$*$-converges in $BV(\Omega)$ to u *if $(u_n)_n$ converges to u in $L^1(\Omega)$ and $(Du_n)_n$ weakly-$*$-converges to Du in Ω, i.e.,*

$$\lim_{n\to+\infty} \int_\Omega \phi dDu_n = \int_\Omega \phi dDu \quad \textit{for all } \phi \in C_0(\Omega).$$

A simple criterion for weak-$*$-convergence is stated in the following result:

Proposition 2.6.7. *Let $(u_n)_n \subset BV(\Omega)$. Then $(u_n)_n$ weakly-$*$-converges to u in $BV(\Omega)$ if and only if $(u_n)_n$ is bounded in $BV(\Omega)$ and converges to u in $L^1(\Omega)$.*

For BV functions the following useful compactness theorem can be stated:

Theorem 2.6.8. *Every sequence $(u_n)_n \subset BV(\Omega)$ satisfying*

$$\sup\left\{ \int_E |u_n| dx + |Du_n|(E) : n \in \mathbb{N} \right\} < \infty \quad \textit{for all } E \subset\subset \Omega \textit{ open}$$

admits a subsequence $(u_{n(k)})_k$ converging in $L^1_{loc}(\Omega)$ to $u \in BV(\Omega)$. If Ω is a bounded extension domain and the sequence is bounded in $BV(\Omega)$, then we can say that $u \in BV(\Omega)$ and that the subsequence weakly-$*$-converges to u.

Let $\Omega_1 \cup \Omega_2 \subset \Omega \subset \bar\Omega_1 \cup \bar\Omega_2$ be a disjoint decomposition of Ω. Moreover we denote by u_{Ω_1} the restriction of u to Ω_1. Then the total variation has the following splitting property, cf. [2, Theorem 3.84, p 177],

$$|D(u_{\Omega_1}+u_{\Omega_2})|(\Omega) = |Du_{\Omega_1}|(\Omega_1)+|Du_{\Omega_2}|(\Omega_2)+\underbrace{\int_{\partial\Omega_1\cap\partial\Omega_2} |u^+_{\Omega_1}(x) - u^-_{\Omega_2}(x)| d\mathcal{H}_{N-1}(x)}_{\text{additional interface term}}, \tag{2.6}$$

where $\mathcal{H}_N$ denotes the Hausdorff measure of dimension N, see Definition 2.4.5. The symbols u^+ and u^- define the "interior" trace of u and the "exterior" trace of u on $\partial\Omega_1 \cap \partial\Omega_2$.

2.7 On Γ-Convergence: Discrete to Continuous Approximation

As it is our intention to develop iterative algorithms based on domain decompositions, which minimize energies including total variation terms, and to prove their convergence by using compactness arguments, we need to work with a topology which is strong enough for traces of functions on boundaries to be continuous. Unfortunately, while the trace of bounded variation functions on boundaries of Lipschitz domains is indeed continuous with respect to the topology induced by the BV-norm (2.4), it is not continuous with respect to the weak-$*$-topology, which is the natural one associated to bounded sequences, see Theorem 2.6.8. Hence most of our results of convergence will be developed in a discretized setting, which is finite dimensional, and has only strong topology. From an applicative point of view this is not a restriction, as in practice, in order to implement such algorithms we need to discretize the problem. Moreover, by using the notion of Γ-convergence, which was originally introduced by De Giorgi [49, 50] with the aim of giving a meaning to the convergence of a sequence of functionals, we show that discretized versions of the total variation in fact Γ-converge to the continuous one. This implies that discrete minimizers of functionals with total variation terms are appropriate approximations in the weak-$*$-sense of minimizers of the continuous functionals.

Definition 2.7.1. *Let (X, d) be a metric space and $F, F_n : X \to \bar{\mathbb{R}}$. We say that a sequence of functions $(F_n)_n$* Γ-converges *to F if:*

(i) For every $u \in X$ and for every sequence $(u_n)_n \subset X$ converging to u we have

$$F(u) \leq \liminf_{n \to +\infty} F_n(u_n).$$

(ii) For every $u \in X$ there exists a sequence $(u_n)_n \subset X$ converging to u such that

$$F(u) \geq \limsup_{n \to +\infty} F_n(u_n).$$

One important consequence of Definition 2.7.1 is that subject to lower semicontinuity and coercivity conditions, if a sequence of functionals F_n Γ-converges to the target functional F, then the corresponding minimizers of F_n also converge to minimizers of F, see [48, Corollary 7.20].

Now we would like to discuss the concept of Γ-convergence for the total variation in more detail. For practical use often a discretization of the total variation is considered. In order to prove that a discretization of the total variation indeed Γ-converges to the continuous total variation, one actually needs only standard concepts from the literature. Nevertheless, the standard literature, as for instance [12], does not completely cover this concrete case. Therefore we make an effort to discuss in details in $\mathbb{R}$ and $\mathbb{R}^2$ the Γ-convergence properties of a discretization of the total variation.

Example 1: Let $\Omega = [0,1]$, $u : \Omega \to \mathbb{R}$ and $F : L^1([0,1]) \to \mathbb{R}$ given by

$$F(u) = \begin{cases} |Du|([0,1]) & \text{if } u \in BV([0,1]) \\ +\infty & \text{otherwise} \end{cases}.$$

Divide Ω into $n \in \mathbb{N} \setminus \{0\}$ equidistant intervals $[\frac{i}{n}, \frac{i+1}{n}]$, $i = 1, \ldots, n-1$. Moreover we define the set of piecewise affine functions

$$\mathcal{A}_n([0,1]) = \{u \in W^{1,1}([0,1]) : u \text{ is affine on each interval } [x_i, x_{i+1}] \subset [0,1], \\ i = 1, \ldots, n-1\}.$$

We set $x_i = \frac{i}{n}$ and we use the notation $u_i = u(x_i)$. If $u \in \mathcal{A}_n([0,1])$, then we also have that the derivative of u denoted by u' is

$$u' = \frac{u_{i+1} - u_i}{1/n} \tag{2.7}$$

on the interval $[x_i, x_{i+1}]$. We study now the limit of the sequence (G_n) as $n \to +\infty$ with $G_n : \mathcal{A}_n([0,1]) \to [0, +\infty]$ given by

$$G_n(u) = \sum_{i=0}^{n-1} |u_{i+1} - u_i| \,.$$

Note that G_n may be interpreted as the discrete analogue of the continuous total variation. The "integral counterpart" of G_n is given by

$$F_n(u) = \begin{cases} \int_0^1 |u'| dx & \text{if } u \in \mathcal{A}_n([0,1]) \\ +\infty & \text{otherwise} \end{cases}, \tag{2.8}$$

where u' is as in (2.7). With this identification we have that G_n Γ-converges to F if F_n defined in (2.8) Γ-converges to F. Hence we need to check now the conditions (i) and (ii) of Definition 2.7.1:

(i) For every $u \in L^1([0,1])$ and for every sequence $(u_n)_n$ converging to u in $L^1(0,1)$ we have
$$F(u) \leq \liminf_{n \to +\infty} F_n(u_n).$$
This immediately follows from the lower semicontinuity of the total variation with respect to the L^1-convergence and hence of the lower semicontinuity of F, i.e., $\liminf_{n\to+\infty} F(u_n) \geq F(u)$.

(ii) For every $u \in L^1([0,1])$ there exists a sequence $(u_n)_n$ converging to u in $L^1([0,1])$ such that
$$F(u) \geq \limsup_{n \to +\infty} F_n(u_n).$$

Note that for $u \in L^1([0,1]) \setminus BV([0,1])$ we have that $F(u) = +\infty$ and hence any sequence $(u_n)_n \subset L^1([0,1])$ converging to u has the property

$$\limsup_{n\to+\infty} F_n(u_n) \leq F(u) = +\infty.$$

Thus without loss of generality we assume that $u \in BV(0,1)$.

For $u \in BV([0,1])$ by Theorem 2.6.5 there exists a sequence $(v_n)_n \subset C^\infty([0,1])$ converging to u in $L^1([0,1])$ such that

$$\lim_{n\to+\infty} \int_0^1 |\nabla v_n| dx = |Du|([0,1]).$$

Moreover assume that $u_n \in \mathcal{A}_n([0,1])$ for all n such that $u_n(x_i) = v_n(x_i)$. By noting that u_n' is constant on each interval $[x_i, x_{i+1}]$ for all $i = 0, \dots, n-1$ we obtain

$$\begin{aligned} F_n(u_n) &= \int_0^1 |u_n'(x)|dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} |u_n'(x)|dx = \sum_{i=0}^{n-1} |\int_{x_i}^{x_{i+1}} u_n'(x)dx| \\ &= \sum_{i=0}^{n-1} |u_n(x_i) - u_n(x_{i+1})| = \sum_{i=0}^{n-1} |v_n(x_i) - v_n(x_{i+1})| \qquad (2.9) \\ &\leq \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} |v_n'(x)|dx = \int_0^1 |v_n'(x)|dx. \end{aligned}$$

Now we take the lim sup for $n \to +\infty$ on both sides in (2.9) and by Theorem 2.6.5 we obtain

$$\limsup_{n\to+\infty} F_n(u_n) \leq \limsup_{n\to+\infty} \int_0^1 |v_n'(x)|dx = F(u).$$

Now it is left to show that the sequence $(u_n)_n$ indeed converges to $u \in BV([0,1])$ in $L^1([0,1])$ or equivalently that $(u_n - v_n)_n$ converges to 0 in $L^1([0,1])$. We have

$$\begin{aligned} \int_0^1 |u_n(x) - v_n(x)|dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} |u_n(x) - v_n(x)|dx \\ &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} \left| \int_{x_i}^{x} u_n'(t) - v_n'(t)dt \right| dx \\ &\leq \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} dx \int_{x_i}^{x_{i+1}} |u_n'(t) - v_n'(t)|\, dt \\ &= \frac{1}{n} \int_0^1 |u_n'(t) - v_n'(t)|\, dt. \end{aligned}$$

Now by the triangle inequality and the observation that there exists a constant $c < +\infty$ such that $\|v_n'\|_{L^1([0,1])} \leq c$ and $\|u_n'\|_{L^1([0,1])} \leq c$ we obtain

$$\frac{1}{n}\int_0^1 |u_n(x) - v_n(x)|dx \leq \frac{1}{n}\left(\int_0^1 |u_n'(t)|dt + \int_0^1 |v_n'(t)|dt\right) \leq \frac{2c}{n}.$$

If we take in the latter inequality the limit for $n \to +\infty$, then we obtain that $\lim_{n\to+\infty} \|u_n - v_n\|_{L^1([0,1])} = 0$. Hence we showed that the discrete functional E_n indeed Γ-converges to F.

Example 2: Let $h_n = 2^{-n}$ for $n \in \mathbb{N} \setminus \{0\}$ and we consider a regular mesh of triangles $\mathcal{T}_{h_n}$ with nodes on the regular grid $h_n \cdot \mathbb{Z} \cap [0,1]^2$.

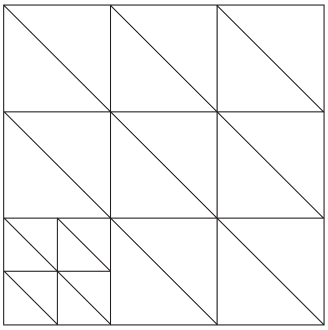


Figure 2.1: *Dyadic mesh of triangles $\mathcal{T}_{h_n}$ with nested finer mesh.*

Note that the triangular mesh $\mathcal{T}_{h_{n+1}}$ is nested in the coarser mesh $\mathcal{T}_{h_n}$, see Figure 2.1. Associated to this mesh we consider the following space

$$\mathcal{A}_{h_n} = \{u \in C([0,1]^2) : u_{|_\tau} \text{ is an affine function for all triangles } \tau \in \mathcal{T}_{h_n}\}.$$

Note that $u_{|_\tau}(x) = \langle x, c_\tau\rangle_{\mathbb{R}^2} + \alpha_\tau$ where $c_\tau \in \mathbb{R}^2$ and $\alpha_\tau \in \mathbb{R}$. In particular $\nabla u_{|_\tau} = c_\tau \in \mathbb{R}^2$.
Every node of the mesh is denoted by $x_{i,j} = (h_n \cdot i, h_n \cdot j)$ for $i, j = 0, \ldots, h_n^{-1} = 2^n$. With this notation we write for $u \in \mathcal{A}_{h_n}$

$$u_{i,j} := u(x_{i,j}).$$

Every two triangles $\tau_{i,j}^-$ and $\tau_{i,j}^+$ that share a long side form a square, see Figure 2.2.

It is not difficult to show that

$$\nabla u_{|_{\tau^-}} = \left(\frac{u_{i+1,j} - u_{i,j}}{h_n}, \frac{u_{i,j+1} - u_{i,j}}{h_n}\right)$$

and

$$\nabla u_{|_{\tau^+}} = \left(\frac{u_{i+1,j+1} - u_{i,j+1}}{h_n}, \frac{u_{i+1,j+1} - u_{i+1,j}}{h_n}\right)$$

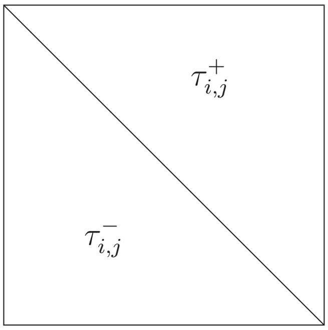


Figure 2.2: *Square $Q_{i,j}$ formed by two triangles, i.e., $Q_{i,j} = \tau^-_{i,j} \cup \tau^+_{i,j}$.*

for all $u \in \mathcal{A}_{h_n}$. Hence, if we denote $Q_{i,j} = \tau^-_{i,j} \cup \tau^+_{i,j}$, we have

$$\begin{aligned}\int_{Q_{i,j}} |\nabla u(x)|dx &= \int_{\tau^-_{i,j}} \sqrt{\left|\frac{u_{i+1,j}-u_{i,j}}{h_n}\right|^2 + \left|\frac{u_{i,j+1}-u_{i,j}}{h_n}\right|^2} dx \\ &\quad + \int_{\tau^+_{i,j}} \sqrt{\left|\frac{u_{i+1,j+1}-u_{i,j+1}}{h_n}\right|^2 + \left|\frac{u_{i+1,j+1}-u_{i+1,j}}{h_n}\right|^2} dx \\ &= \frac{h_n^2}{2}\Big(\sqrt{\left|\frac{u_{i+1,j}-u_{i,j}}{h_n}\right|^2 + \left|\frac{u_{i,j+1}-u_{i,j}}{h_n}\right|^2} \\ &\quad + \sqrt{\left|\frac{u_{i+1,j+1}-u_{i,j+1}}{h_n}\right|^2 + \left|\frac{u_{i+1,j+1}-u_{i+1,j}}{h_n}\right|^2}\Big) \\ &= \frac{h_n}{2}\Big(\sqrt{|u_{i+1,j}-u_{i,j}|^2 + |u_{i,j+1}-u_{i,j}|^2} \\ &\quad + \sqrt{|u_{i+1,j+1}-u_{i,j+1}|^2 + |u_{i+1,j+1}-u_{i+1,j}|^2}\Big).\end{aligned}$$

Hence

$$\begin{aligned}\int_{[0,1]^2} |\nabla u(x)|dx = \frac{h_n}{2}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1} \Big(&\sqrt{|u_{i+1,j}-u_{i,j}|^2 + |u_{i,j+1}-u_{i,j}|^2} \\ &+ \sqrt{|u_{i+1,j+1}-u_{i,j+1}|^2 + |u_{i+1,j+1}-u_{i+1,j}|^2}\Big).\end{aligned} \tag{2.10}$$

We have two functionals in $L^1([0,1]^2)$:

$$F(u) = \begin{cases} |Du|([0,1]^2) & \text{if } u \in BV([0,1]^2) \\ +\infty & \text{otherwise} \end{cases}$$

$$F_{h_n}(u) = \begin{cases} \int_{[0,1]^2} |\nabla u(x)|dx & \text{if } u \in \mathcal{A}_{h_n} \\ +\infty & \text{otherwise} \end{cases}.$$

We need to prove the following two conditions, in order to show that F_{h_n} Γ-converges to F:

(i) For every $u \in L^1([0,1]^2)$ and for every sequence $(u_n)_n \subset L^1([0,1]^2)$ converging to u such that
$$F(u) \leq \liminf_{n\to+\infty} F_{h_n}(u_n).$$
This condition comes for free by the lower semicontinuity of the total variation with respect to the L^1-convergence.

(ii) For every $u \in L^1([0,1]^2)$ there exists a sequence $(u_n)_n \subset L^1([0,1]^2)$ converging to u such that
$$F(u) \geq \limsup_{n\to+\infty} F_{h_n}(u_n).$$
Actually we construct a sequence $(u_n)_n$ such that $\limsup_{n\to+\infty} F_{h_n}(u_n) = F(u)$.

Note that for $u \in L^1([0,1]^2) \setminus BV([0,1]^2)$, we have that $F(u) = +\infty$. Hence any sequence $(u_n)_n \subset L^1([0,1]^2)$ converging to u has the property
$$F_{h_n}(u_n) \leq F(u) = +\infty$$
and hence $\limsup_{n\to+\infty} F_{h_n}(u_n) \leq F(u) = +\infty$. Therefore without loss of generality we assume $u \in BV([0,1]^2)$.

For $u \in BV([0,1]^2)$ by Theorem 2.6.5 there exists a sequence $(v_n)_n$ in $C^\infty([0,1]^2)$ such that $v_n \to u$ in $L^1([0,1]^2)$ and
$$\lim_{n\to+\infty} \int_{[0,1]^2} |\nabla v_n| dx = |Du|([0,1]^2). \tag{2.11}$$

For all $v \in C^\infty([0,1]^2)$ we are able to construct an interpolating piecewise affine function $v_{h_n} \in \mathcal{A}_{h_n}$, i.e., $v(x_{i,j}) = v_{h_n}(x_{i,j})$ for all $i,j = 0,\ldots,2^n$. We define the operator
$$\Pi_{h_n} : C^\infty([0,1]^2) \to \mathcal{A}_{h_n}, \quad v \mapsto v_{h_n} = \Pi_{h_n} v.$$

By [34, Theorem 3.1.5] we have
$$\Pi_{h_n} v \to v \quad \text{in } L^1([0,1]^2) \text{ for } n \to +\infty \quad \text{for all } v \in C^\infty([0,1]^2)$$
and
$$\int_{[0,1]^2} |\nabla v - \nabla \Pi_{h_n} v| dx \leq c h_n \int_{[0,1]^2} |\nabla^2 v| dx. \tag{2.12}$$

Hence also $\nabla \Pi_{h_n} v \to \nabla v$ in $L^1([0,1]^2)$ as $n \to +\infty$. We have that for all $k \in \mathbb{N} \setminus \{0\}$ there exists an $n_k > n_{k-1}$ such that for all $n \geq n_k$
$$\left| |Du|([0,1]^2) - \int_{[0,1]^2} |\nabla v_n| dx \right| \leq 2^{-k-1},$$

which is due to (2.11), and we have that there exists an $\tilde{n} = \tilde{n}(n)$ such that

$$\left| \int_{[0,1]^2} |\nabla v_n| dx - \int_{[0,1]^2} |\nabla \Pi_{h_{\tilde{n}}} v_n| dx \right| \leq 2^{-k-1},$$

because of (2.12). Hence by the triangle inequality we obtain that

$$\left| |Du|([0,1]^2) - \int_{[0,1]^2} |\nabla \Pi_{h_{\tilde{n}}} v_n| dx \right| \leq 2^{-k}.$$

Actually through n we have $\tilde{n} = \tilde{n}(n_k) = \tilde{n}_k$.

Therefore we can construct a sequence

$$\Pi_{h_{\tilde{n}_k}} v_{n_k} \in \mathcal{A}_{h_{\tilde{n}_k}}$$

such that

$$\lim_{k \to +\infty} \underbrace{\int_{[0,1]^2} |\nabla \Pi_{h_{\tilde{n}_k}} v_{n_k}| dx}_{= F_{h_{\tilde{n}_k}}(u_{\tilde{n}_k})} = |Du|([0,1]^2).$$

Without loss of generality we assume that $\tilde{n}_{k+1} > \tilde{n}_k$. Notice now that

$$\Pi_{h_{\tilde{n}_k}} v_{n_k} \in \mathcal{A}_{h_n} \quad \text{for all } n \geq \tilde{n}_k,$$

and this is due to the nesting property of finer meshes, in particular for $\tilde{n}_k \leq n \leq \tilde{n}_{k+1} - 1$. Hence, up to repeating the same term

$$u_n = \Pi_{h_{\tilde{n}_k}} v_{n_k} \quad \text{for all } n = \tilde{n}_k, \ldots, \tilde{n}_{k+1} - 1,$$

we constructed a sequence $u_n \in \mathcal{A}_{h_n}$ such that $u_n \to u$ in $L^1([0,1]^2)$ and

$$\lim_{n \to +\infty} F_{h_n}(u_n) = F(u).$$

This concludes the proof of the second condition and shows that F_{h_n} indeed Γ-converges to F.

Remark 2.7.2. *The definition of discrete total variation which we use later for our discretization, see Section 3.2.2, slightly differs from* (2.10)*. This choice is motivated by simpler computation of the divergences (see* (3.23)*), which is used heavily in the numerical implementation, see Section 3.2.6 (e.g., Chambolle's algorithm in* (3.63)*). Nevertheless, the two definitions are nearly equivalent and in practice they produce similar minimizers.*

Chapter 3

Subspace Correction for Non-smooth and Non-additive Problems

In this chapter we introduce a subspace correction method for functionals involving a non-smooth and non-additive term. Section 3.1 is based on the work [66], where a subspace correction method for such problems is proposed. There an implementation of this algorithm is suggested that is guaranteed to converge and to decrease the objective energy monotonically. Only under technical conditions on the interfaces of the subspaces it is possible to prove its convergence to minimizers of $\mathcal{J}$. We will show a simple counterexample, see Remark 3.1.1, which emphasizes that the convergence to an expected minimizer of subspace correction methods for non-smooth and non-additive problems is far from being obvious. Moreover in Section 3.2, which is based on the work [63], we specify the subspace correction algorithm from [66] to the case of an orthogonal wavelet decomposition and for deblurring problems. This provides us with a framework in which we can construct another more specific and more sophisticated counterexample, which shows that in general also for such splittings we cannot expect convergence to a minimizer of $\mathcal{J}$, even for the simplest case of the identity operator $T = I$. Further we show additional properties of the limit of the sequence produced by the algorithm, and obtain an additional condition under which the obtained limit is indeed the expected minimizer. Although this newly obtained condition cannot be ensured to hold always for any operator T, we show that an orthogonal wavelet space decomposition for deblurring problems works in practice very efficiently, as already observed by Vonesch and Unser in their study related to ℓ_1-regularization [116]. In particular, with the help of the newly obtained condition of convergence, we are able to show in our numerical examples that the sequence produced by this algorithm in fact numerically converges to a minimizer of $\mathcal{J}$.

3.1 An Alternating Algorithm for Orthogonal Splittings

In the sequel $\mathcal{H}$ is a real separable Hilbert space endowed with the norm $\|\cdot\|_{\mathcal{H}}$. Let us recall the subspace correction method proposed by Fornasier and Schönlieb in [66]. We are interested in the minimization in $\mathcal{H}$ of functionals in the general form

$$\mathcal{J}(u) := \|Tu - g\|_{\mathcal{H}}^2 + 2\alpha\psi(u), \tag{3.1}$$

where $T : \mathcal{H} \to \mathcal{H}$ is a bounded linear operator, $g \in \mathcal{H}$ is a datum, $\alpha > 0$ is a fixed constant, and $\psi : \mathcal{H} \to \mathbb{R}^+ \cup \{+\infty\}$ is a sublinear, 1-homogeneous and lower semicontinuous function. For such ψ there exists a closed convex set $K_\psi \subset \mathcal{H}$ such that

$$\psi^*(u) = \sup_{v \in \mathcal{H}} \{\langle u, v\rangle - \psi(v)\} = \chi_{K_\psi}(u) = \begin{cases} 0 & \text{if } u \in K_\psi \\ +\infty & \text{otherwise} \end{cases}.$$

We assume in the following that $K_\psi = -K_\psi$. Moreover for any convex set $K \subset \mathcal{H}$ we denote $P_K(u) = \arg\min_{v \in K} \|u - v\|_{\mathcal{H}}$ the orthogonal projection onto K. Assume that associated with ψ there exists a dense subspace $\mathcal{H}^\psi$ of $\mathcal{H}$ endowed with the norm

$$\|u\|_{\mathcal{H}^\psi} := \|u\|_{\mathcal{H}} + \psi(u).$$

Since $\mathcal{H}^\psi$ is a dense subspace of $\mathcal{H}$ we have that

$$\mathcal{H}^\psi \subset \mathcal{H} \simeq \mathcal{H}^* \subset (\mathcal{H}^\psi)^*,$$

where the symbole $\simeq$ denotes an isomorphism and the duality $\langle\cdot,\cdot\rangle_{(\mathcal{H}^\psi)^* \times \mathcal{H}^\psi}$ extends the scalar product on $\mathcal{H}$. In particular, $\mathcal{H}$ is weakly-$*$-dense in $(\mathcal{H}^\psi)^*$. Further we require that

$(H1)$ bounded subsets in $\mathcal{H}^\psi$ are sequentially bounded in another topology τ^ψ of $\mathcal{H}^\psi$;

$(H2)$ ψ is lower-semicontinuous with respect to the topology τ^ψ;

$(H3)$ $\mathcal{H}^\psi = \{u \in \mathcal{H} : \psi(u) < \infty\}$.

In order to guarantee the existence of minimizers for $\mathcal{J}$ we assume that:

$$\mathcal{J} \text{ is coercive in } \mathcal{H}, \text{ i.e., } \{\mathcal{J} \le C\} := \{u \in \mathcal{H} : \mathcal{J}(u) \le C\} \text{ is bounded in } \mathcal{H}. \tag{\textbf{C}}$$

Instead of minimizing $\mathcal{J}$ on the space $\mathcal{H}$ we decompose $\mathcal{H}$ into two mutually orthogonal, and complementary subspaces V_1, V_2, i.e., $\mathcal{H} = V_1 \oplus V_2$. By $\pi_{V_i} : \mathcal{H} \to V_i$ we define the corresponding orthogonal projections onto V_i, for $i = 1, 2$. Then every

$u \in \mathcal{H}$ has a unique representation $u = \pi_{V_1}(u) + \pi_{V_2}(u)$. In the sequel we denote $u_i = \pi_{V_i}(u)$ for $i = 1, 2$. Further we require the mapping property

$$\pi_{V_i}|_{\mathcal{H}^\psi} : \mathcal{H}^\psi \to V_i^\psi := \mathcal{H}^\psi \cap V_i, \quad i = 1, 2,$$

continuously in the norm of $\mathcal{H}^\psi$, and $\operatorname{Range}(\pi_{V_i}|_{\mathcal{H}^\psi}) = V_i^\psi$ is closed. This implies that $\mathcal{H}^\psi$ splits into the direct sum $\mathcal{H}^\psi = V_1^\psi \oplus V_2^\psi$. We are mainly interested in the situation where for such an orthogonal splitting we have that

$$\psi(\pi_{V_1}(u) + \pi_{V_2}(v)) \neq \psi(\pi_{V_1}(u)) + \psi(\pi_{V_2}(v)),$$

as it happens in the case of the total variation, see (2.6). For simplicity, in this thesis we principally restrict ourself to split the problem into two subspaces V_1, V_2, by noting that one can easily generalize the argument to a decomposition into multiple subspaces $V_1, \dots, V_{\mathcal{N}}$, cf. Remark 4.2.10, and we provide several numerical examples in this context, see Section 4.2.4 and Section 4.4.3.

With this splitting we want to minimize $\mathcal{J}$ by suitable instances of the following alternating algorithm: pick an initial $V_1 \oplus V_2 \ni u_1^{(0)} + u_2^{(0)} := u^{(0)} \in \mathcal{H}^\psi$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} u_1^{(n+1)} \approx \arg\min_{v_1 \in V_1} \mathcal{J}(v_1 + u_2^{(n)}) \\ u_2^{(n+1)} \approx \arg\min_{v_2 \in V_2} \mathcal{J}(u_1^{(n+1)} + v_2) \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)}. \end{cases} \tag{3.2}$$

We use "≈" (the approximation symbol) because in practice we never perform the exact minimization. Before we discuss in the next section how to realize the approximation to the individual subspace minimizations we introduce a counterexample from [118] for a non-smooth and non-additive function on $[0, 1]^2$, for which such a splitting method in general does not work.

Remark 3.1.1 (Counterexample). *Let $V_1 = V_2 = [0, 1]$, $\mathcal{H} = V_1 \times V_2$ and*

$$\mathcal{J}(u) = \mathcal{J}(u_1 + u_2) = \mathcal{J}(u_1, u_2) = \begin{cases} u_1 - 2u_2 & \text{if } 0 \le u_2 \le u_1 \le 1 \\ u_2 - 2u_1 & \text{if } 0 \le u_1 \le u_2 \le 1 \end{cases}.$$

We observe that $\mathcal{J}(u) = |u_1 - u_2| - \min(u_1, u_2)$ and hence it is a convex function on $\mathcal{H}$. Moreover the minimizer of $\min_{u_2} \mathcal{J}(u_1, u_2)$ and $\min_{u_1} \mathcal{J}(u_1, u_2)$ are $u_1 = u_2$. Thus $u_1^{(1)} = u_2^{(0)}, u_2^{(1)} = u_1^{(1)} = u_2^{(0)}$ and hence for all $\ell \ge 1$, $u_1^{(\ell)} = u_2^{(\ell)} = u_2^{(0)}$ and $\mathcal{J}(u^{(\ell)}) = -u_2^{(0)}$, whereas $\min_{u \in \mathcal{H}} \mathcal{J}(u) = -1$.

3.1.1 Subspace Minimization

Let us consider, for example, the subspace minimization of (3.2) in V_1:

$$\arg\min_{u_1 \in V_1} \mathcal{J}(u_1 + u_2) = \arg\min_{u_1 \in V_1} \|Tu_1 - (g - Tu_2)\|_{\mathcal{H}}^2 + 2\alpha\psi(u_1 + u_2). \tag{3.3}$$

First of all, observe that $\{u \in \mathcal{H} : \pi_{V_2} u = u_2, \mathcal{J}(u) \leq C\} \subset \{\mathcal{J} \leq C\}$, hence the former set is also bounded by assumption (**C**) and the minimization (3.3) has solutions.

In order to realize the approximation to the individual subspace minimization, it is useful for us to introduce an auxiliary functional $\mathcal{J}_1^s$, called the *surrogate functional* of $\mathcal{J}$: assume $a, u_1 \in V_1$ and $u_2 \in V_2$ and define

$$\mathcal{J}_1^s(u_1 + u_2, a) := \mathcal{J}(u_1 + u_2) + \|u_1 - a\|_{\mathcal{H}}^2 - \|T(u_1 - a)\|_{\mathcal{H}}^2. \tag{3.4}$$

A straightforward computation shows that

$$\mathcal{J}_1^s(u_1 + u_2, a) = \|u_1 - (a + \pi_{V_1} T^*(g - Tu_2 - Ta))\|_{\mathcal{H}}^2 + 2\alpha\psi(u_1 + u_2) + \Phi(a, g, u_2),$$

where Φ is a function of a, g, u_2 only. Note that now the variable u_1 is not anymore affected by the action of T. Consequently, we want to realize an approximate solution to (3.3) by using the following algorithm: for $u_1^{(0)} \in V_1^{\psi}$,

$$u_1^{(\ell+1)} = \arg\min_{u_1 \in V_1} \mathcal{J}_1^s(u_1 + u_2, u_1^{(\ell)}), \quad \ell \geq 0. \tag{3.5}$$

Before proving the convergence of this algorithm, we need to clarify first how to practically compute $u_1^{(\ell+1)}$ for $u_1^{(\ell)}$ given. To this end we need to introduce further notions and to recall some useful results.

Remark 3.1.2. *Assume $\varphi \geq 0$ is a proper lower semicontinuous convex function. For $F(u; z) = \|u - z\|_{\mathcal{H}}^2 + 2\varphi(u)$, we define the function*

$$\operatorname{prox}_{\varphi}(z) := \arg\min_{u \in \mathcal{H}} F(u; z),$$

which is called the proximity map *in the convex analysis literature, e.g., [53, 39], and* generalized thresholding *in the signal processing literature, e.g., [42, 44, 45, 61]. Observe that by $\varphi \geq 0$ the function F is coercive in $\mathcal{H}$ and by lower semicontinuity and strict convexity of the term $\|u - z\|_{\mathcal{H}}^2$ this definition is well-posed. In particular,* $\operatorname{prox}_{\varphi}(z)$ *is the unique solution of the following differential inclusion*

$$0 \in (u - z) + \partial\varphi(u).$$

It is well-known [53, 102] that the proximity map is nonexpansive, i.e.,

$$\|\operatorname{prox}_{\varphi}(z_1) - \operatorname{prox}_{\varphi}(z_2)\|_{\mathcal{H}} \leq \|z_1 - z_2\|_{\mathcal{H}} \quad \textit{for all } z_1, z_2 \in \mathcal{H}.$$

In particular, if φ is a 1-homogeneous function then

$$\operatorname{prox}_{\varphi}(z) = (I - P_{K_{\varphi}})(z),$$

where K_{φ} is a suitable closed convex set associated to φ, see for instance [39].

We realize that in order to address the subminimization problem

$$\arg\min_{u_1 \in V_1} \mathcal{J}_1^s(u_1+u_2,a) = \arg\min_{u_1 \in V_1} \|u_1 - z_1\|_{\mathcal{H}}^2 + 2\alpha\psi(u_1+u_2),$$

where $z_1 = a + \pi_{V_1} T^*(g - Tu_2 - Ta)$, we have to solve a constrained optimization problem of the type

$$\arg\min_{x \in \mathcal{H}} \{F(x) : G(x) = 0\}, \tag{3.6}$$

where $F : \mathcal{H} \to \mathbb{R}$ is a convex function and $G : \mathcal{H} \to \mathcal{H}$ is a bounded linear operator on $\mathcal{H}$. There exist a variety of methods that solve this type of constrained minimization problems, as the Augmented Lagrangian Method [72] and its adaptations known under the name of the Bregman iterations [19, 20, 69, 96, 97, 122, 123, 124] because of their relationship to Bregman distance [14]. Here, for simplicity, we use the *iterative oblique thresholding* algorithm as proposed in the work [66], which is based on the following useful result:

Theorem 3.1.3 (Generalized Lagrange multipliers for non-smooth objective functions, Theorem 1.8, [6])**.** *If F is continuous in a point of $\ker G$ and G has closed range in $\mathcal{H}$, then a point $x_0 \in \ker G$ is an optimal solution of (3.6) if and only if*

$$\partial F(x_0) \cap \operatorname{Range} G^* \neq \emptyset.$$

By using this result one can show the following statement that provides a method to solve each iteration step of (3.5).

Theorem 3.1.4 (Oblique thresholding)**.** *For $u_2 \in V_2^\psi$ and for $z \in V_1$ the following statements are equivalent:*

(i) $u_1^* = \arg\min_{u \in V_1} \|u - z\|_{\mathcal{H}}^2 + 2\alpha\psi(u+u_2)$;

(ii) there exists $\eta \in \operatorname{Range}(\pi_{V_2}|_{\mathcal{H}^\psi})^* \simeq (V_2^\psi)^*$ *such that* $0 \in u_1^* - (z-\eta) + \alpha\partial_{\mathcal{H}^\psi}\psi(u_1^* + u_2)$.

Moreover, the following statements are equivalent and imply (i) and (ii).

(iii) there exists $\eta \in V_2$ such that $u_1^* = (I - P_{\alpha K_\psi})(z + u_2 - \eta) - u_2$;

(iv) there exists $\eta \in V_2$ such that $\eta = \pi_{V_2} P_{\alpha K_\psi}(\eta - (z+u_2))$.

Proof. Let us show the equivalence between (i) and (ii). The problem in (i) can be reformulated as

$$u_1^* = \arg\min_{u \in \mathcal{H}^\psi} \{F(u) := \|u - z\|_{\mathcal{H}}^2 + 2\alpha\psi(u+u_2), \pi_{V_2}(u) = 0\}.$$

The latter is a special instance of (3.6). Moreover, F is continuous on $V_1^\psi \subset V_1 = \ker \pi_{V_2}$ in the norm-topology of $\mathcal{H}^\psi$ (while in general it is not on V_1 with the norm topology of $\mathcal{H}$). Recall now that $\pi_{V_2}|_{\mathcal{H}^\psi}$ is assumed to be a bounded and surjective map with closed range in the norm-topology of $\mathcal{H}^\psi$ (see above). This means that $(\pi_{V_2}|_{\mathcal{H}^\psi})^*$ is injective and that $\text{Range}(\pi_{V_2}|_{\mathcal{H}^\psi})^* \simeq (V_2^\psi)^*$ is closed. Therefore, by an application of Theorem 3.1.3 the optimality of u_1^* is equivalent to the existence of $\eta \in \text{Range}(\pi_{V_2}|_{\mathcal{H}^\psi})^* \simeq (V_2^\psi)^*$ such that

$$-\eta \in \partial_{\mathcal{H}^\psi} F(u_1^*).$$

Due to the continuity of $\|u - z\|_{\mathcal{H}}^2$ in $\mathcal{H}^\psi$, we have, by Proposition 2.3.4, that

$$\partial_{\mathcal{H}^\psi} F(u_1^*) = 2(u_1^* - z) + 2\alpha \partial_{\mathcal{H}^\psi} \psi(u_1^* + u_2).$$

Thus, the optimality of u_1^* is equivalent to

$$0 \in u_1^* - (z - \eta) + \alpha \partial_{\mathcal{H}^\psi} \psi(u_1^* + u_2).$$

This concludes the equivalence of (i) and (ii). Let us show now that (iii) implies (ii). The condition in (iii) can be rewritten as

$$\xi = (I - P_{\alpha K_\psi})(z + u_2 - \eta), \quad \xi = u_1^* + u_2.$$

Since $\psi \geq 0$ is 1-homogeneous and lower semincontinuous, by Remark 3.1.2, the latter is equivalent to

$$0 \in \xi - (z + u_2 - \eta) + \alpha \partial_{\mathcal{H}} \psi(\xi)$$

or, by (H3),

$$\begin{aligned} \xi &= \arg\min_{u \in \mathcal{H}} \|u - (z + u_2 - \eta)\|_{\mathcal{H}}^2 + 2\alpha\psi(u) \\ &= \arg\min_{u \in \mathcal{H}^\psi} \|u - (z + u_2 - \eta)\|_{\mathcal{H}}^2 + 2\alpha\psi(u) \end{aligned}$$

The latter optimal problem is equivalent to

$$0 \in \xi - (z + u_2 - \eta) + \alpha \partial_{\mathcal{H}^\psi} \psi(\xi) \text{ or } 0 \in u_1^* - (z - \eta) + \alpha \partial_{\mathcal{H}^\psi} \psi(u_1^* + u_2).$$

Since $V_2 \subset (V_2^\psi)^* \simeq \text{Range}(\pi_{V_2}|_{\mathcal{H}^\psi})^*$ we obtain that (iii) implies (ii). We prove now the equivalence between (iii) and (iv). We have

$$\begin{aligned} u_1^* &= (I - P_{\alpha K_\psi})(z + u_2 - \eta) - u_2 \in V_1 \\ &= z - \eta - P_{\alpha K_\psi}(z + u_2 - \eta). \end{aligned}$$

By applying π_{V_2} to both sides of the latter equality we get

$$0 = -\eta - \pi_{V_2} P_{\alpha K_\psi}(z + u_2 - \eta).$$

By recalling that $K_\psi = -K_\psi$, we obtain the fixed point equation

$$\eta = \pi_{V_2} P_{\alpha K_\psi}(\eta - (z + u_2)). \tag{3.7}$$

Conversely, assume $\eta = \pi_{V_2} P_{\alpha K_\psi}(\eta - (z + u_2))$ for some $\eta \in V_2$. Then

$$\begin{aligned}(I - P_{\alpha K_\psi})(z + u_2 - \eta) - u_2 &= z - \eta - P_{\alpha K_\psi}(z + u_2 - \eta)\\ &= z - \pi_{V_2} P_{\alpha K_\psi}(\eta - (z + u_2)) - P_{\alpha K_\psi}(z + u_2 - \eta)\\ &= z - (I - \pi_{V_2}) P_{\alpha K_\psi}(z + u_2 - \eta)\\ &= z - \pi_{V_1} P_{\alpha K_\psi}(z + u_2 - \eta) = u_1^* \in V_1.\end{aligned}$$

□

Remark 3.1.5. *For $\mathcal{H}$ of finite dimension, which is the relevant case in numerical applications, all the spaces are independent of the particular attached norm and coincide with their duals. Hence all the statements (i)-(iv) of the previous theorem are equivalent in this case.*

We wonder now whether any of the conditions in Theorem 3.1.4 is indeed practically satisfied. In particular, we want to show that $\eta \in V_1$ as in (iii) or (iv) of the previous theorem is provided as the limit of the following iterative algorithm:

$$\eta^{(0)} \in V_2, \quad \eta^{(m+1)} = \pi_{V_2} P_{\alpha K_\psi}(\eta^{(m)} - (z + u_2)), \quad m \geq 0. \tag{3.8}$$

Proposition 3.1.6. *The following statements are equivalent:*

(i) there exists $\eta \in V_2$ such that $\eta = \pi_{V_2} P_{\alpha K_\psi}(\eta - (z + u_2))$ (which is in turn the condition (iv) of Theorem 3.1.4)

(ii) the iteration (3.8) converges weakly to any $\eta \in V_2$ that satisfies (3.7).

In particular, there are no fixed points of (3.7) if and only if $\|\eta^{(m)}\|_{\mathcal{H}} \to \infty$, for $m \to \infty$.

For the proof of this proposition we need to recall some well-known notions and results.

Definition 3.1.7. *A nonexpansive map $T : \mathcal{H} \to \mathcal{H}$ is strongly nonexpansive if for $(u_n - v_n)_n$ bounded and $\|T(u_n) - T(v_n)\|_{\mathcal{H}} - \|u_n - v_n\|_{\mathcal{H}} \to 0$ we have*

$$u_n - v_n - T(u_n) - T(v_n) \to 0, \quad n \to \infty.$$

Proposition 3.1.8 (Corollaries 1.3, 1.4, and 1.5 [15])**.** *Let $T : \mathcal{H} \to \mathcal{H}$ be a strongly nonexpansive map. Then $\operatorname{fix} T = \{u \in \mathcal{H} : T(u) = u\} \neq \emptyset$ if and only if $(T^n u)_n$ converges weakly to a fixed point $u_0 \in \operatorname{fix} T$ for any choice of $u \in \mathcal{H}$.*

Proof. (Proposition 3.1.6) Orthogonal projections onto convex sets are strongly nonexpansive [7, Corollary 4.2.3]. Moreover, compositions of strongly nonexpansive maps are strongly nonexpansive [15, Lemma 2.1]. By an application of Proposition 3.1.8 we immediately have the result, since any map of the type $T(\xi) = Q(\xi) + \xi_0$ is strongly nonexpansive whenever Q is (this is a simple observation from the definition of strongly nonexpansive map). Indeed, we are looking for fixed points of $\eta = \pi_{V_2} P_{\alpha K_\psi}(\eta - (z + u_2))$ or, equivalently, of $\xi = \underbrace{\pi_{V_2} P_{\alpha K_\psi}}_{:=Q}(\xi) - \underbrace{(z + u_2)}_{:=\xi_0}$. □

Convergence of the Subspace Minimization

For $u_1^* = \arg\min_{u_1 \in V_1} \|u_1 - z\| + 2\alpha\psi(u_1 + u_2)$ we observed by Remark 3.1.2 that

$$u_1^* = \mathrm{prox}_{\alpha\psi(\cdot+u_2)}(z).$$

Now let us denote

$$\mathbb{S}_\alpha^{\psi,V_1,V_2}(z;u_2) := \mathrm{prox}_{\alpha\psi(\cdot+u_2)}(z).$$

From [39, Subsection 2.3] we can conclude that

$$\|\mathbb{S}_\alpha^{\psi,V_1,V_2}(z_1;u_2) - \mathbb{S}_\alpha^{\psi,V_1,V_2}(z_2;u_2)\|_{\mathcal{H}} \le \|z_1 - z_2\|_{\mathcal{H}}, \quad \text{for all } z_1, z_2 \in V_1.$$

In view of the previous results it was observed in [66] that the iteration (3.5) can be computed by

$$u_1^{(\ell+1)} = \mathbb{S}_\alpha^{\psi,V_1,V_2}(u_1^{(\ell)} + \pi_{V_1}T^*(g - Tu_2 - Tu_1^{(\ell)}); u_2). \tag{3.9}$$

In certain cases, e.g., in finite dimensions, this iteration can be explicitly expressed as

$$u_1^{(\ell+1)} = \mathbb{S}_\alpha^{\psi}(u_1^{(\ell)} + \pi_{V_1}T^*(g - Tu_2 - Tu_1^{(\ell)}) + u_2 - \eta^{(\ell)}) - u_2,$$

where $\mathbb{S}_\alpha^\psi = (I - P_{\alpha K_\psi})$ and $\eta^{(\ell)} \in V_2$ is any solution of the fixed point equation

$$\eta = \pi_{V_2}P_{\alpha K_\psi}(\eta - (u_1^{(\ell)} + \pi_{V_1}T^*(g - Tu_2 - Tu_1^{(\ell)}) + u_2)).$$

The computation of $\eta^{(\ell)}$ can be implemented by the algorithm (3.8).

Theorem 3.1.9. *Assume $u_2 \in V_2^\psi$ and $\|T\| < 1$. Then the iteration (3.9) converges weakly to a solution $u_1^* \in V_1^\psi$ of (3.3) for any initial choice of $u_1^{(0)} \in V_1^\psi$.*

Proof. For the sake of completeness, we report the proof of this theorem, which follows the same strategy already proposed in the paper [42]. Similar results can also be found in [39]. In particular we want to apply Opial's fixed point theorem [95]:

Theorem 3.1.10. *Let the mapping A from $\mathcal{H}$ to $\mathcal{H}$ satisfy the following conditions:*

(i) A is nonexpansive: for all $z, z' \in \mathcal{H}$, $\|Az - Az'\|_{\mathcal{H}} \le \|z - z'\|_{\mathcal{H}}$;

(ii) A is asymptotically regular: for all $z \in \mathcal{H}$, $\|A^{n+1}z - A^n z\|_{\mathcal{H}} \to 0$, for $n \to +\infty$;

(iii) the set $\mathcal{F} = \mathrm{fix}\, A$ of fixed points of A in $\mathcal{H}$ is not empty.

Then for all $z \in \mathcal{H}$, the sequence $(A^n z)_n$ converges weakly to a fixed point in $\mathcal{F}$.

We need to prove that $A(u_1) := \mathbb{S}_\alpha^{\psi,V_1,V_2}(u_1 + \pi_{V_1} T^*(g - Tu_2 - Tu_1); u_2)$ fulfills the assumptions of Theorem 3.1.10 on V_1.

Step 1. As stated at the beginning of this section, there exist solutions $u_1^* \in V_1^\psi$ to (3.3). With a similar argument to the one used to prove the equivalence of (i) and (ii) in Theorem 3.1.4, the optimality of u_1^* can be proven to be equivalent to

$$0 \in -\pi_{V_1} T^*(g - Tu_2 - Tu_1^*) + \eta + \alpha \partial_{\mathcal{H}^\psi} \psi(u_1^* + u_2),$$

for some $\eta \in (V_2^\psi)^*$. By adding and subtracting u_1^* we obtain

$$0 \in u_1^* - (\underbrace{(u_1^* + \pi_{V_1} T^*(g - Tu_2 - Tu_1^*))}_{:=z} - \eta) + \alpha \partial_{\mathcal{H}^\psi} \psi(u_1^* + u_2).$$

By applying the equivalence of (i) and (ii) in Theorem 3.1.4 we obtain that u_1^* is a fixed point of the following equation

$$u_1^* = \mathbb{S}_\alpha^{\psi,V_1,V_2}(u_1^* + \pi_{V_1} T^*(g - Tu_2 - Tu_1^*); u_2),$$

hence $\operatorname{fix} A \neq \emptyset$.

Step 2. The algorithm produces iterations that are asymptotically regular, i.e., $\|u_1^{(\ell+1)} - u_1^{(\ell)}\|_{\mathcal{H}} \to 0$. Observe that, for $a \in V_i$ and $\|T\| < 1$,

$$\|u_i - a\|_{\mathcal{H}}^2 - \|Tu_i - Ta\|_{\mathcal{H}}^2 \geq C \|u_i - a\|_{\mathcal{H}}^2, \tag{3.10}$$

for $C = (1 - \|T\|^2) > 0$. Hence

$$\mathcal{J}(u) = \mathcal{J}_i^s(u, u_i) \leq \mathcal{J}_i^s(u, a), \tag{3.11}$$

and

$$\mathcal{J}_i^s(u, a) - \mathcal{J}_i^s(u, u_i) \geq C \|u_i - a\|_{\mathcal{H}}^2. \tag{3.12}$$

Then we have the following estimates

$$\begin{aligned} \mathcal{J}(u_1^{(\ell)} + u_2) &= \mathcal{J}_1^s(u_1^{(\ell)} + u_2, u_1^{(\ell)}) \\ &\geq \mathcal{J}_1^s(u_1^{(\ell+1)} + u_2, u_1^{(\ell)}) \\ &\geq \mathcal{J}_1^s(u_1^{(\ell+1)} + u_2, u_1^{(\ell+1)}) = \mathcal{J}(u_1^{(\ell+1)} + u_2). \end{aligned}$$

Since $(\mathcal{J}(u_1^{(\ell)} + u_2))_\ell$ is monotonically decreasing and bounded from below by 0, necessarily it is a convergent sequence. Moreover,

$$\mathcal{J}(u_1^{(\ell)} + u_2) - \mathcal{J}(u_1^{(\ell+1)} + u_2) \geq C \|u_1^{(\ell+1)} - u_1^{(\ell)}\|_{\mathcal{H}}^2,$$

and the latter convergence implies $\|u_1^{(\ell+1)} - u_1^{(\ell)}\|_{\mathcal{H}} \to 0$.

Step 3. We are left with showing the nonexpansiveness of A. By the nonexpansiveness of $\mathbb{S}_\alpha^{\psi,V_1,V_2}(\cdot; u_2)$ we obtain

$$\begin{aligned} &\|\mathbb{S}_\alpha^{\psi,V_1,V_2}(u_1^1 + \pi_{V_1} T^*(g - Tu_2 - Tu_1^1; u_2) - \mathbb{S}_\alpha^{\psi,V_1,V_2}(u_1^2 + \pi_{V_1} T^*(g - Tu_2 - Tu_1^2; u_2)\|_{\mathcal{H}} \\ &\leq \|u_1^1 + \pi_{V_1} T^*(g - Tu_2 - Tu_1^1) - (u_1^2 + \pi_{V_1} T^*(g - Tu_2 - Tu_1^2)\|_{\mathcal{H}} \\ &= \|(I - \pi_{V_1} T^* T \pi_{V_1})(u_1^1 - u_1^2)\|_{\mathcal{H}} \\ &\leq \|u_1^1 - u_1^2\|_{\mathcal{H}}. \end{aligned}$$

In the latter inequality we used once more that $\|T\| < 1$. □

Remark 3.1.11. *Note that the condition $\|T\| < 1$ in Theorem 3.1.9, and assumed also later, is not automatically satisfied in general. However, in case the norm of the given operator T exceeds 1, a proper rescaling of the problem re-establishes the desired setting. The rescaling can be easily obtained by multiplying the functional $\mathcal{J}$ by a positive constant $\gamma < \frac{1}{\|T\|^2}$ and by minimizing the resulting functional*

$$\mathcal{J}_\gamma(u) = \|\sqrt{\gamma}Tu - \sqrt{\gamma}g\|_2^2 + 2\gamma\alpha|Du|(\Omega).$$

Note that the minimizers of the rescaled problem do coincide with minimizers of $\mathcal{J}$.

Let us conclude this subsection by mentioning that all the results presented here hold symmetrically for the minimization on V_2, and that the notations should be just adjusted accordingly.

3.1.2 Convergence of the Sequential Subspace Correction Method

The suggested algorithm in [66] is to solve

$$\arg\min_{u\in\mathcal{H}} \mathcal{J}(u)$$

by picking an initial $V_1 \oplus V_2 \ni u_1^{(0,L)} + u_2^{(0,M)} := u^{(0)} \in \mathcal{H}^\psi$, e.g., $u_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = u_1^{(n,L)} \\ u_1^{(n+1,\ell+1)} = \arg\min_{u_1\in V_1} \mathcal{J}_1^s(u_1 + u_2^{(n,M)}, u_1^{(n+1,\ell)}) \quad \ell = 0, \dots, L-1 \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = u_2^{(n,M)} \\ u_2^{(n+1,m+1)} = \arg\min_{u_2\in V_2} \mathcal{J}_2^s(u_1^{(n+1,L)} + u_2, u_2^{(n+1,m)}) \quad m = 0, \dots, M-1 \end{cases} \\ u^{(n+1)} := u_1^{(n+1,L)} + u_2^{(n+1,M)}. \end{cases} \tag{3.13}$$

Now let us recall the main convergence results established in [66] for any choice of finite numbers L and M of inner iterations.

Theorem 3.1.12 (Convergence properties)**.** *The algorithm in (3.13) produces a sequence $(u^{(n)})_n$ in $\mathcal{H}^\psi$ with the following properties:*

(i) $\mathcal{J}(u^{(n)}) > \mathcal{J}(u^{(n+1)})$ for all $n \in \mathbb{N}$ (unless $u^{(n)} = u^{(n+1)}$);

(ii) $\lim_{n\to\infty} \|u^{(n+1)} - u^{(n)}\|_{\mathcal{H}} = 0$;

(iii) the sequence $(u^{(n)})_n$ has subsequences which converge weakly in $\mathcal{H}$ and in $\mathcal{H}^\psi$ endowed with the topology τ^ψ;

(iv) if we additionally assume, for simplicity, that $\dim \mathcal{H} < \infty$, $(u^{(n_k)})_k$ *is a strongly converging subsequence, and* $u^{(\infty)}$ *is its limit, then* $u^{(\infty)}$ *is a minimizer of* $\mathcal{J}$ *whenever one of the following conditions holds*

(a) $\psi(u_1^{(\infty)} + \eta_2) + \psi(u_2^{(\infty)} + \eta_1) - \psi(u_1^{(\infty)} + u_2^{(\infty)}) \le \psi(\eta_1 + \eta_2)$ *for all* $\eta_i \in V_i$, $i = 1, 2$;

(b) ψ *is differentiable at* $u^{(\infty)}$ *with respect to* V_i *for one* $i \in \{1, 2\}$, *i.e., there exists* $\frac{\partial}{\partial V_i}\psi(u^{(\infty)}) := \zeta_i \in (V_i)^*$ *such that*

$$\langle \zeta_i, v_i \rangle = \lim_{t \to 0} \frac{\psi(u_1^{(\infty)} + u_2^{(\infty)} + tv_i) - \psi(u_1^{(\infty)} + u_2^{(\infty)})}{t}, \text{ for all } v_i \in V_i.$$

Proof. Let us first observe that

$$\mathcal{J}(u^{(n)}) = \mathcal{J}_1^s(u_1^{(n)} + u_2^{(n)}, u_1^{(n)}) \quad = \quad \mathcal{J}_1^s(u_1^{(n,L)} + u_2^{(n)}, u_1^{(n+1,0)}).$$

From the definition of $u_1^{(n+1,1)}$ and from the minimal properties of $u_1^{(n+1,1)}$ in (3.13) we have

$$\mathcal{J}_1^s(u_1^{(n,L)} + u_2^{(n)}, u_1^{(n+1,0)}) \ge \mathcal{J}^s(u_1^{(n+1,1)} + u_2^{(n)}, u_1^{(n+1,0)}).$$

From (3.11) we have

$$\mathcal{J}_1^s(u_1^{(n+1,1)} + u_2^{(n)}, u_1^{(n+1,0)}) \ge \mathcal{J}_1^s(u_1^{(n+1,1)} + u_2^{(n)}, u_1^{(n+1,1)}).$$

Putting in line these inequalities we obtain

$$\mathcal{J}(u^{(n)}) \ge \mathcal{J}(u_1^{(n+1,1)} + u_2^{(n)}).$$

In particular, from (3.12) we have

$$\mathcal{J}(u^{(n)}) - \mathcal{J}(u_1^{(n+1,1)} + u_2^{(n)}) \ge C\|u_1^{(n+1,1)} - u_1^{(n+1,0)}\|_{\mathcal{H}}^2.$$

After L steps we conclude the estimate

$$\mathcal{J}(u^{(n)}) \quad \ge \quad \mathcal{J}(u_1^{(n+1,L)} + u_2^{(n)}),$$

and

$$\mathcal{J}(u^{(n)}) - \mathcal{J}(u_1^{(n+1,L)} + u_2^{(n)}) \ge C \sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_{\mathcal{H}}^2.$$

By definition of $u_2^{(n+1,1)}$ and its minimal properties we have

$$\mathcal{J}(u_1^{(n+1,L)} + u_2^{(n)}) \ge \mathcal{J}_2^s(u_1^{(n+1,L)} + u_2^{(n+1,1)}, u_2^{(n+1,0)}).$$

By similar arguments as above we finally find the decreasing estimate

$$\mathcal{J}(u^{(n)}) \ge \mathcal{J}_2^s(u_1^{(n+1,L)} + u_2^{(n+1,M)}) = \mathcal{J}(u^{(n+1)}), \tag{3.14}$$

and

$$\mathcal{J}(u^{(n)}) - \mathcal{J}(u^{(n+1)})$$
$$\geq C \left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_{\mathcal{H}}^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_{\mathcal{H}}^2 \right). \tag{3.15}$$

From (3.14) we have $\mathcal{J}(u^{(0)}) \geq \mathcal{J}(u^{(n)})$. By the coerciveness condition (**C**), $(u^{(n)})_n$ is uniformly bounded in $\mathcal{H}^\psi$, hence there exists a $\mathcal{H}$-weakly- and τ^ψ-convergent subsequence $(u^{(n_j)})_j$. Let us denote $u^{(\infty)}$ the weak limit of the subsequence. For simplicity, we rename such a subsequence by $(u^{(n)})_n$. Moreover, since the sequence $(\mathcal{J}(u^{(n)}))_n$ is monotonically decreasing and bounded from below by 0, it is also convergent. From (3.15) and the latter convergence we deduce

$$\left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_{\mathcal{H}}^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_{\mathcal{H}}^2 \right) \to 0, \quad n \to \infty. \tag{3.16}$$

In particular, by the standard inequality $(a^2+b^2) \geq \frac{1}{2}(a+b)^2$ for $a, b > 0$ and the triangle inequality, we have also

$$\|u^{(n)} - u^{(n+1)}\|_{\mathcal{H}} \to 0, \quad n \to \infty. \tag{3.17}$$

We would like now to show that the following outer lower semicontinuity holds

$$0 \in \lim_{n\to\infty} \partial \mathcal{J}(u^{(n)}) \subset \partial \mathcal{J}(u^{(\infty)}).$$

For this we need to assume that $\mathcal{H}$-weakly- and $\tau^\psi-$convergences do imply strong convergence in $\mathcal{H}$. This is the case, e.g., when $\dim(\mathcal{H}) < \infty$. The optimality condition for $u_1^{(n+1,L)}$ is equivalent to

$$0 \in u_1^{(n+1,L)} - z_1^{(n+1)} + \alpha \partial_{V_1} \psi(\cdot + u_2^{(n,M)})(u_1^{(n+1,L)}),$$

where

$$z_1^{(n+1)} := u_1^{(n+1,L-1)} + \pi_{V_1} T^*(g - Tu_2^{(n,M)} - Tu_1^{(n+1,L-1)}).$$

Analogously we have

$$0 \in u_2^{(n+1,M)} - z_2^{(n+1)} + \alpha \partial_{V_2} \psi(\cdot + u_1^{(n+1,L)})(u_2^{(n+1,M)}),$$

where

$$z_2^{(n+1)} := u_2^{(n+1,M-1)} + \pi_{V_2} T^*(g - Tu_1^{(n+1,L)} - Tu_2^{(n+1,M-1)}).$$

Due to the strong convergence of the sequence $u^{(n)}$ and by (3.16) we have the following limits for $n \to \infty$

$$\xi_1^{(n+1)} := u_1^{(n+1,L)} - z_1^{(n+1)} \to \xi_1 := -\pi_{V_1} T^*(g - Tu_2^{(\infty)} - Tu_1^{(\infty)}) \in V_1,$$

$$\xi_2^{(n+1)} := u_2^{(n+1,M)} - z_2^{(n+1)} \to \xi_2 := -\pi_{V_2} T^*(g - Tu_2^{(\infty)} - Tu_1^{(\infty)}) \in V_2,$$

and

$$\xi_1^{(n+1)} + \xi_2^{(n+1)} \to \xi := T^*(Tu^{(\infty)} - g).$$

Moreover, we have

$$-\frac{1}{\alpha}\xi_1^{(n+1)} \in \partial_{V_1}\psi(\cdot + u_2^{(n,M)})(u_1^{(n+1,L)}),$$

meaning that

$$\langle -\frac{1}{\alpha}\xi_1^{(n+1)}, \eta_1 - u_1^{(n+1,L)}\rangle + \psi(u_1^{(n+1,L)} + u_2^{(n,M)}) \leq \psi(\eta_1 + u_2^{(n,M)}), \quad \text{for all } \eta_1 \in V_1.$$

Analogously we have

$$\langle -\frac{1}{\alpha}\xi_2^{(n+1)}, \eta_2 - u_2^{(n+1,M)}\rangle + \psi(u_1^{(n+1,L)} + u_2^{(n+1,M)}) \leq \psi(\eta_2 + u_1^{(n+1,L)}), \quad \text{for all } \eta_2 \in V_2.$$

By taking the limits for $n \to \infty$ and by (3.16) we obtain

$$\langle -\frac{1}{\alpha}\xi_1, \eta_1 - u_1^{(\infty)}\rangle + \psi(u^{(\infty)}) \leq \psi(\eta_1 + u_2^{(\infty)}), \quad \text{for all } \eta_1 \in V_1 \tag{3.18}$$

$$\langle -\frac{1}{\alpha}\xi_2, \eta_2 - u_2^{(\infty)}\rangle + \psi(u^{(\infty)}) \leq \psi(\eta_2 + u_1^{(\infty)}), \quad \text{for all } \eta_2 \in V_2. \tag{3.19}$$

These latter conditions are rewritten in vector form as

$$0 \in \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \alpha \left(\partial_{V_1}\psi(\cdot + u_2^{(\infty)})(u_1^{(\infty)}) \times \partial_{V_2}\psi(\cdot + u_1^{(\infty)})(u_2^{(\infty)}) \right). \tag{3.20}$$

Observe now that

$$2\xi + 2\alpha\partial_{\mathcal{H}}\psi(u^{(\infty)}) = 2T^*(Tu^{(\infty)} - g) + 2\alpha\partial_{\mathcal{H}}\psi(u^{(\infty)}) = \partial\mathcal{J}(u^{(\infty)}).$$

If $0 \in \xi + \alpha\partial_{\mathcal{H}}\psi(u^{(\infty)})$, then we would have the wanted minimality condition. While the inclusion

$$\partial_{\mathcal{H}}\psi(u^{(\infty)}) \subset \partial_{V_1}\psi(\cdot + u_2^{(\infty)})(u_1^{(\infty)}) \times \partial_{V_2}\psi(\cdot + u_1^{(\infty)})(u_2^{(\infty)}),$$

easily follows from the definition of a subdifferential, the converse inclusion, which would imply from (3.20) the wished minimality condition, does not hold in general. Thus, we show the converse inclusion under one of the following two conditions:

(a) $\psi(u_1^{(\infty)} + \eta_2) + \psi(u_2^{(\infty)} + \eta_1) - \psi(u_1^{(\infty)} + u_2^{(\infty)}) \leq \psi(\eta_1 + \eta_2)$ for all $\eta_i \in V_i$, $i = 1, 2$;

(b) ψ is differentiable at $u^{(\infty)}$ with respect to V_i for one $i \in \{1, 2\}$, i.e., there exists $\frac{\partial}{\partial V_i}\psi(u^{(\infty)}) := \zeta_i \in (V_i)^*$ such that

$$\langle \zeta_i, v_i \rangle = \lim_{t \to 0} \frac{\psi(u_1^{(\infty)} + u_2^{(\infty)} + tv_i) - \psi(u_1^{(\infty)} + u_2^{(\infty)})}{t}, \text{ for all } v_i \in V_i.$$

Let us start with condition (a). We want to show that

$$\langle -\frac{1}{\alpha}\xi, \eta - u^{(\infty)}\rangle + \psi(u^{(\infty)}) \leq \psi(\eta), \quad \text{for all } \eta \in \mathcal{H},$$

or, equivalently, that

$$\langle -\frac{1}{\alpha}\xi_1, \eta_1 - u_1^{(\infty)}\rangle + \langle -\frac{1}{\alpha}\xi_2, \eta_2 - u_2^{(\infty)}\rangle + \psi(u_1^{(\infty)} + u_2^{(\infty)}) \leq \psi(\eta_1 + \eta_2), \quad \text{for all } \eta_i \in V_i,$$

By the differential inclusions (3.18) and (3.19) we have

$$\langle -\frac{1}{\alpha}\xi_1, \eta_1 - u_1^{(\infty)}\rangle + \langle -\frac{1}{\alpha}\xi_2, \eta_2 - u_2^{(\infty)}\rangle + 2\psi(u_1^{(\infty)} + u_2^{(\infty)}) \leq \psi(u_1^{(\infty)} + \eta_2) + \psi(u_2^{(\infty)} + \eta_1),$$

for all $\eta_i \in V_i$, hence

$$\begin{aligned} &\langle -\frac{1}{\alpha}\xi_1, \eta_1 - u_1^{(\infty)}\rangle + \langle -\frac{1}{\alpha}\xi_2, \eta_2 - u_2^{(\infty)}\rangle + \psi(u_1^{(\infty)} + u_2^{(\infty)}) \\ &\qquad \leq \psi(u_1^{(\infty)} + \eta_2) + \psi(u_2^{(\infty)} + \eta_1) - \psi(u_1^{(\infty)} + u_2^{(\infty)}), \quad \text{for all } \eta_i \in V_i. \end{aligned}$$

An application of condition (a) concludes the proof of the wanted differential inclusion.

Let us show the inclusion now under the assumption of condition (b). Without loss of generality, we assume that ψ is differentiable at $u^{(\infty)}$ with respect to V_2. First of all we define $\tilde{\psi}(u_1, u_2) := \psi(u_1 + u_2)$. Since ψ is convex, by an application of [102, Corollary 10.11], we have

$$\begin{aligned} \partial_{V_1}\psi(\cdot + u_2)(u_1) \simeq \partial_{u_1}\tilde{\psi}(u_1, u_2) = \{\zeta_1 \in V_1^* : \exists \zeta_2 \in V_2^* : \\ (\zeta_1, \zeta_2)^T \in \partial\tilde{\psi}(u_1, u_2) \simeq \partial_{\mathcal{H}}\psi(u_1 + u_2)\}. \end{aligned}$$

Since ψ is differentiable at $u^{(\infty)}$ with respect to V_2, for any $(\zeta_1, \zeta_2)^T \in \partial\tilde{\psi}(u_1, u_2) \simeq \partial_{\mathcal{H}}\psi(u_1 + u_2)$ we have necessarily $\zeta_2 = \frac{\partial}{\partial V_2}\psi(u^{(\infty)})$ as the unique member of $\partial_{V_2}\psi(\cdot + u_1^{(\infty)})(u_2^{(\infty)})$. Hence, the following inclusion must also hold

$$\begin{aligned} 0 &\in \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \alpha\left(\partial_{V_1}\psi(\cdot + u_2^{(\infty)})(u_1^{(\infty}) \times \partial_{V_2}\psi(\cdot + u_1^{(\infty)})(u_2^{(\infty)})\right) \\ &\subset \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} + \alpha\partial_{V_1 \times V_2}\tilde{\psi}(u_1, u_2) \\ &\simeq \xi + \alpha\partial_{\mathcal{H}}\psi(u^{(\infty)}). \end{aligned}$$

□

This shows that the algorithm in (3.13) converges and decreases the energy $\mathcal{J}$ monotonically. Only under the technical condition (a) or (b) of the previous theorem, which are in general not fulfilled, the convergence to minimizers of the original functional (3.1) could be ensured.

3.2 Counterexample for Wavelet Decomposition

In this section we specify the previously introduced subspace correction method to the case of an orthogonal wavelet space decomposition and for deblurring problems. These specifications allow us to construct another more specific counterexample (cf. Proposition 3.2.5) to show that convergence of the algorithm in (3.2) cannot be obtained in general for non-smooth and non-additive problems, now for $\psi(u) = |Du|(\Omega)$ with Ω being an open bounded set.

The successful use of wavelet-based multilevel algorithms for deblurring problems where the ℓ_1-norm of the wavelet coefficients is used as a regularization term, cf. (1.21), was shown in [116]. There algorithm (1.23) was used with minor modifications, specifically by using Haar wavelets for deblurring (or deconvolution) problems, where cyclic updates of the different resolution levels were combined with the preconditioning effect of subband-specific parameters. The effectiveness of this method was shown by solving multidimensional image deconvolution problems in 3D fluorescence microscopy. We give a brief and intuitive explanation of the reason why this multilevel method works so well for deblurring problems: wavelet space decompositions split the function space into orthogonal subspaces V_i. Note that now $T : L^2(\Omega) \to L^2(\Omega)$ is just a convolution operator with kernel κ, i.e., $Tu = \kappa * u$, or a multiplier $\hat{\kappa}$ in the Fourier domain, where the V_i's represent nearly disjoint dyadic subbands, and we have that all T_{V_i} are also nearly orthogonal, i.e., $T^*_{V_i} T_{V_{\hat{i}}} \approx 0$ for $i \neq \hat{i}$, where T_{V_i} denotes the operator T restricted to the subspace V_i. Hence each subiteration (1.24) of the algorithm in (1.23) is (nearly) restricted to one of the V_i, independent of other subiterations, and converges fast because $T^*_{V_i} T_{V_i}$ is a well-conditioned operator. This is the case whenever the Fourier transform $\hat{\kappa}$ is, for example, a slowly decaying function on the subband associated with V_i, see Figure 3.1.

To gain maximal performance of the algorithm in (1.23) we need to introduce preconditioner constants for each subiteration respectively, i.e., instead of considering $I - T^*_{V_i} T_{V_i}$ we take iteration operators

$$I - \frac{1}{\alpha_i} T^*_{V_i} T_{V_i},$$

for $\alpha_i \geq \|T_{V_i}\|^2$.

The main goal of this chapter is to transpose these observations on preconditioning effects of alternating algorithms based on wavelet decompositions to the deblurring model where the term $\|u_\Lambda\|_{\ell_1(\Lambda)}$ in (1.21) is substituted by the total variation of the function u.

Our reason for expecting that the preconditioning effects observed by Vonesch and Unser [116] for Haar wavelet-based regularization will take place also in total variation regularization of deblurring problems stems from the well-known near characterization of BV in terms of wavelets [37, 38]: the BV-norm of a bivariate

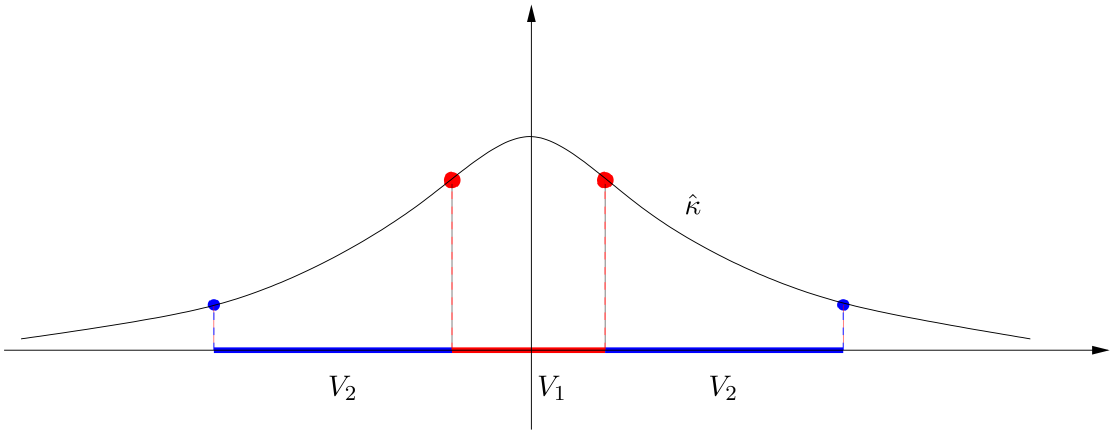


Figure 3.1: *We depict a slowly decaying envelope of the Fourier transform $\hat{\kappa}$ of a kernel κ. The spaces V_1 and V_2 are two orthogonal spaces, obtained by a wavelet decomposition and associated to nearly disjoint subbands. Restricted on the subband associated to V_i, the function $\hat{\kappa}$, essentially representing the spectrum of the matrix T_{V_i}, can be intuitively understood as bounded from above and below, providing the well-conditioning of the operator $T_{V_i}^* T_{V_i}$.*

function u is in fact nearly equivalent to the ℓ_1-norm of its bivariate Haar wavelet coefficients u_Λ. More precisely, there exist constants $c_1, c_2 \in \mathbb{R}^+$ such that

$$c_1 \|u_\Lambda\|_{\ell_{1+\delta}} \leq \|u\|_{L^1(\Omega)} + |Du|(\Omega) \leq c_2 \|u_\Lambda\|_{\ell_1}, \text{ for all } u \in BV(\Omega), \tag{3.21}$$

and for all $\delta > 0$. Actually these inequalities result in embeddings of BV with respect to suitable Besov spaces:

$$B^1_{1,1} \subset BV \subset B^{1,w}_{1,1}.$$

We refer to [38] for more details.

Because of this observation and the above mentioned preconditioning mechanism for a deblurring operator in connection with a wavelet space decomposition, we are interested in the minimization of the functional (1.4) by using a suitably adapted wavelet-based multilevel algorithm.

Let $(\mathcal{V}_i)_{i\in\mathbb{Z}}$ be a multiresolution analysis generated by a scaling function and let ψ be a corresponding wavelet function. Then we obtain

$$L^2(\Omega) = \overline{\bigcup_{i\in\mathbb{Z}} \mathcal{V}_i} = \overline{\mathcal{V}_i \oplus \bigoplus_{j=i}^{\infty} \mathcal{W}_j} = \overline{\bigoplus_{j\in\mathbb{Z}} \mathcal{W}_j},$$

where $\mathcal{W}_j$ is the wavelet space corresponding to the j-th level generated by the basis

$$\{\psi_\lambda : \lambda \in \Lambda_j\},$$

and Λ_j denotes the set of indices for the j-th level, see [36, 41] for more details. Moreover, $\mathcal{W}_j$ is the orthogonal complement of $\mathcal{V}_j$ in $\mathcal{V}_{j+1}$, i.e., we have

$$\mathcal{V}_{j+1} = \mathcal{V}_j \oplus \mathcal{W}_j. \tag{3.22}$$

In particular we may decompose $L^2(\Omega)$ in the following way

$$L^2(\Omega) = \mathcal{V}_0 \oplus \mathcal{V}_0^\perp = \mathcal{V}_0 \oplus \overline{\left(\bigoplus_{j=0}^{\infty} \mathcal{W}_j\right)}$$

and denote $V_1 := \mathcal{V}_0$ and $V_2 := \mathcal{V}_0^\perp = \overline{\bigoplus_{j=0}^{\infty} \mathcal{W}_j}$. Associated with this wavelet decomposition into two subspaces the minimization of (1.4) can be carried out by the alternating subspace correction method (3.2).

3.2.1 Technical Issues

In the rest of the thesis we eventually work on a finite dimensional space by considering a finite regular mesh as a discretization of Ω. Thus we consider instead of the continuous functional (1.4) its discrete approximation, for ease again denoted by $\mathcal{J}$ in (3.24). In Section 2.7 we showed that such a discrete approximation Γ-converges to the continuous functional and has the same singular nature as the continuous problem. In particular, we recall, that discrete minimizers of (3.24), interpolated by piecewise linear functions, converge in weak-$*$-topology of BV to minimizers of the functional (1.4) in the continuous setting. Of course, when dealing with numerical solutions, only the discrete approach matters together with its approximation properties to the continuous problem. However, the need of working in the discrete setting is not only practical, it is also topological. In fact bounded sets in BV are (only) weakly-$*$-compact, and this property is fundamental for showing that certain sequences have converging subsequences. Unfortunately, the weak-$*$-topology of BV is "too weak" for our purpose of obtaining conditions under which the limit of the sequence produced by the subspace correction method is indeed the expected minimizer. Note further that in the discrete setting where topological issues are not a concern anymore, also the dimension d can be arbitrary, contrary to the continuous setting where the dimension d has to be linked to boundedness properties of the operator T, see [113, property H2, pag. 134]. For simplicity reasons, as in the previous section, we will limit ourselves to decompose our problem only into two orthogonal subspaces V_1 and V_2, which is also now by no means a restriction, as a generalization to a multiple decomposition is straightforward, see Remark 4.2.10. However, we stress also that in our numerical experiments the beneficial effect of preconditioning does not seem to improve significantly by considering multiple decompositions, see Section 3.2.6.

3.2.2 Notations

Our analysis is performed for a discrete approximation of the continuous functional (1.4). Essentially we approximate functions u by their sampling on a regular grid and their gradient Du by finite differences here denoted by ∇u.

It is sufficient to us to introduce our main notations for a discretization in $[0,1]^d$, with $d \in \mathbb{N}$, only by noting that they can be easily extended to more general spaces in $\mathbb{R}^d$. Since we are interested in a discrete setting we define the *discrete d-orthotope* $\Omega = \{x_1^1 < \ldots < x_{N_1}^1\} \times \ldots \times \{x_1^d < \ldots < x_{N_d}^d\} \subset [0,1]^d$. The considered function spaces are $\mathcal{H} = \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_d}$, where $N_i \in \mathbb{N}$ for $i = 1, \ldots, d$. For $u \in \mathcal{H}$ we write $u = u(x_{\mathbf{i}})_{\mathbf{i} \in \mathcal{I}}$ with

$$\mathcal{I} := \prod_{k=1}^{d} \{1, \ldots, N_k\}$$

and

$$u(x_{\mathbf{i}}) = u(x_{i_1}^1, \ldots, x_{i_d}^d),$$

where $i_k \in \{1, \ldots, N_k\}$ and $(x_{\mathbf{i}})_{\mathbf{i} \in \mathcal{I}} \in \Omega$. Then we endow $\mathcal{H}$ with the norm

$$\|u\|_{\mathcal{H}} = \|u\|_2 = \left(\sum_{\mathbf{i} \in \mathcal{I}} |u(x_{\mathbf{i}})|^2 \right)^{1/2} = \left(\sum_{x \in \Omega} |u(x)|^2 \right)^{1/2}.$$

We define the scalar product of $u, v \in \mathcal{H}$ as

$$\langle u, v \rangle_{\mathcal{H}} = \sum_{\mathbf{i} \in \mathcal{I}} u(x_{\mathbf{i}}) v(x_{\mathbf{i}})$$

and the scalar product of $p, q \in \mathcal{H}^d$ as

$$\langle p, q \rangle_{\mathcal{H}^d} = \sum_{\mathbf{i} \in \mathcal{I}} \langle p(x_{\mathbf{i}}), q(x_{\mathbf{i}}) \rangle_{\mathbb{R}^d}$$

with $\langle y, z \rangle_{\mathbb{R}^d} = \sum_{j=1}^d y_j z_j$ for every $y = (y_1, \ldots, y_d) \in \mathbb{R}^d$ and $z = (z_1, \ldots, z_d) \in \mathbb{R}^d$. We will consider also other norms, in particular

$$\|u\|_p = \left(\sum_{\mathbf{i} \in \mathcal{I}} |u(x_{\mathbf{i}})|^p \right)^{1/p}, \quad 1 \le p < \infty,$$

and

$$\|u\|_\infty = \sup_{\mathbf{i} \in \mathcal{I}} |u(x_{\mathbf{i}})|.$$

We denote the discrete gradient ∇u by

$$(\nabla u)(x_{\mathbf{i}}) = ((\nabla u)^1(x_{\mathbf{i}}), \ldots, (\nabla u)^d(x_{\mathbf{i}}))$$

with

$$(\nabla u)^j(x_{\mathbf{i}}) = \begin{cases} u(x^1_{i_1}, \dots, x^j_{i_j+1}, \dots, x^d_{i_d}) - u(x^1_{i_1}, \dots, x^j_{i_j}, \dots, x^d_{i_d}) & \text{if } i_j < N_j \\ 0 & \text{if } i_j = N_j \end{cases}$$

for all $j = 1, \dots, d$ and for all $\mathbf{i} = (i_1, \dots, i_d) \in \mathcal{I}$.
Let $\varphi : \mathbb{R} \to \mathbb{R}$, we define for $\omega \in \mathcal{H}^d$

$$\varphi(|\omega|)(\Omega) = \sum_{\mathbf{i}\in\mathcal{I}} \varphi(|\omega(x_{\mathbf{i}})|) = \sum_{x\in\Omega} \varphi(|\omega(x)|),$$

where $|y| = \sqrt{y_1^2 + \ldots + y_d^2}$. In particular we define the *total variation* of u by setting $\varphi(s) = s$ and $\omega = \nabla u$, i.e.,

$$|\nabla u|(\Omega) := \sum_{\mathbf{i}\in\mathcal{I}} |\nabla u(x_{\mathbf{i}})| = \sum_{x\in\Omega} |\nabla u(x)|.$$

Further we introduce the *discrete divergence* $\operatorname{div} : \mathcal{H}^d \to \mathcal{H}$ defined, in analogy with the continuous setting, by $\operatorname{div} = -\nabla^*$ (∇^* is the adjoint of the gradient ∇). The discrete divergence operator is explicitly given by

$$\begin{aligned} (\operatorname{div} p)(x_{\mathbf{i}}) = &\begin{cases} p^1(x^1_{i_1}, \dots, x^d_{i_d}) - p^1(x^1_{i_1-1}, \dots, x^d_{i_d}) & \text{if } 1 < i_1 < N_1 \\ p^1(x^1_{i_1}, \dots, x^d_{i_d}) & \text{if } i_1 = 1 \\ -p^1(x^1_{i_1-1}, \dots, x^d_{i_d}) & \text{if } i_1 = N_1 \end{cases} \\ &+ \ldots + \begin{cases} p^d(x^1_{i_1}, \dots, x^d_{i_d}) - p^d(x^1_{i_1}, \dots, x^d_{i_d-1}) & \text{if } 1 < i_d < N_d \\ p^d(x^1_{i_1}, \dots, x^d_{i_d}) & \text{if } i_d = 1 \\ -p^d(x^1_{i_1}, \dots, x^d_{i_d-1}) & \text{if } i_d = N_d, \end{cases} \end{aligned} \tag{3.23}$$

for every $p = (p^1, \dots, p^d) \in \mathcal{H}^d$ and for all $\mathbf{i} = (i_1, \dots, i_d) \in \mathcal{I}$. (Note that if we considered discrete domains Ω which are not discrete d-orthotopes, then the definitions of gradient and divergence operators should be adjusted accordingly.) With these notations, we define the closed convex set

$$K := \left\{ \operatorname{div} p : p \in \mathcal{H}^d, |p(x)|_\infty \le 1 \text{ for all } x \in \Omega \right\},$$

where $|p(x)|_\infty = \max\left\{|p^1(x)|, \dots, |p^d(x)|\right\}$, and we denote by $P_K(u) = \arg\min_{v\in K} \|u - v\|_2$ the *orthogonal projection onto* K. We will often use the symbol 1 to indicate the constant vector with entry values 1 and the symbol 1_D to indicate the characteristic function of the domain $D \subset \Omega$.

3.2.3 Description of the Algorithm

Preconditioning

We are interested in solving by the multilevel algorithm in (3.2) the minimization of the discrete functional $\mathcal{J} : \mathcal{H} \to \mathbb{R}$ defined by

$$\mathcal{J}(u) = \|Tu - g\|_2^2 + 2\alpha|\nabla u|(\Omega), \tag{3.24}$$

where $T : \mathcal{H} \to \mathcal{H}$ is a blur operator with kernel κ, $g \in \mathcal{H}$ is a given datum, and $\alpha > 0$ is a fixed regularization parameter. Furthermore, it is convenient for us to assume $\|T\| < 1$, which is not a restriction, since a proper rescaling of the problem yields the desired setting, and it does not change the minimization problem, see Remark 3.1.11. In order to guarantee the existence of minimizers for (3.24) we assume condition (**C**), i.e., that $\mathcal{J}$ is coercive in $\mathcal{H}$. It is well known that if $1 \notin \ker(T)$ then this coercivity condition is satisfied, see [113, Proposition 3.1]. In addition, if T is injective, for instance, if κ is a Gaussian or an *averaging* convolution kernel (see Section 3.2.6), then (3.24) has an unique minimizer.

We can identify $\mathcal{H}$ with the sequences of samples $(u(x))_{x\in\Omega}$ of a function u on $[0,1]^d$, and with $\mathcal{V}_1$, the first scaling space of a multiresolution analysis, by means of the map $(u(x))_{x\in\Omega} \to \sum_{\lambda\in\Lambda_1} u(x_\lambda)\varphi_{1,\lambda}$, where $\varphi_{1,\lambda}$ is a properly dilated scaling function, and $(x_\lambda)_{\lambda\in\Lambda}$ is a suitable rearrangement of the nodes of the mesh Ω. Moreover, by property (3.22), we have the orthogonal splitting $\mathcal{H} = \mathcal{V}_1 = \mathcal{V}_0 \oplus \mathcal{W}_0$. Of course, we may obtain further levels of decomposition

$$\mathcal{H} = \mathcal{V}_j \oplus \left(\bigoplus_{i=j}^{0} \mathcal{W}_i \right) \quad j \in \mathbb{Z}^-.$$

For simplicity we restrict ourselves to a decomposition into two subspaces $V_1 := \mathcal{V}_0$ and $V_2 := \mathcal{W}_0$ only. As in Section 3.1 we introduce *surrogate functionals* on $V_1 \oplus V_2$ for $a \in V_i$ and for $i = 1, 2$ by

$$\mathcal{J}^s_{i,\alpha_i}(u_1 + u_2, a) = \mathcal{J}(u_1 + u_2) + \alpha_i \|u_i - a\|_2^2 - \|T(u_i - a)\|_2^2, \tag{3.25}$$

where α_1, α_2 are positive constants chosen as specified below in order to ensure the convergence of the subminimization iteration

$$u_i^{(n+1,\ell+1)} = \arg\min_{u_i \in V_i} \mathcal{J}^s_{i,\alpha_i}(u_1 + u_2, u_i^{(n+1,\ell)}), \quad \ell > 0, \tag{3.26}$$

to a minimizer of the corresponding subproblem of (3.2), i.e.,

$$\arg\min_{u_i \in V_i} \mathcal{J}(u_1 + u_2),$$

for $i = 1, 2$. Let us further define the synthesis operators $S_1 : \ell_2 \to V_1$ via the orthonormal basis for V_1 and $S_2 : \ell_2 \to V_2$ via the orthonormal basis for V_2. That is $u_1 = S_1(u_{\Lambda_1})$ and $u_2 = S_2(u_{\Lambda_2})$ for $u_{\Lambda_1} = (u_\lambda)_{\lambda\in\Lambda_1}$ denoting the scaling function coefficients and $u_{\Lambda_2} = (u_\lambda)_{\lambda\in\Lambda_2}$ denoting the wavelet coefficients. Since S_1, S_2 are isometries, we know that

$$\|T_{V_i}(u_i - a)\|_2^2 = \|T_{V_i} S_i(u_{\Lambda_i} - a_{\Lambda_i})\|_2^2 \quad \text{and} \quad \|u_i - a\|_2^2 = \|u_{\Lambda_i} - a_{\Lambda_i}\|_{\ell_2}^2,$$

where $a = S_1(a_{\Lambda_1})$ or $a = S_2(a_{\Lambda_2})$. Because of these observations it makes sense to choose

$$1 \geq \alpha_i > \|T_{V_i} S_i\|^2 \tag{3.27}$$

for $i = 1, 2$. Then we obtain

$$\|T_{V_i}(u_i - a)\|_2^2 = \|T_{V_i} S_i(u_{\Lambda_i} - a_{\Lambda_i})\|_2^2 \leq \|T_{V_i} S_i\|^2 \|u_{\Lambda_i} - a_{\Lambda_i}\|_{\ell_2}^2 < \alpha_i \|u_i - a\|_2^2.$$

Notice that with constants α_i as in (3.27), we have for $n = 1, 2, \ldots,$

$$\begin{aligned} \mathcal{J}(u^{(n)}) &\leq \mathcal{J}^s_{2,\alpha_2}(u_1^{(n,L)} + u_2^{(n,M)}, u_2^{(n-1,M)}) \leq \mathcal{J}(u_1^{(n,L)} + u_2^{(n-1,M)}) \\ &\leq \mathcal{J}^s_{1,\alpha_1}(u_1^{(n,L)} + u_2^{(n-1,M)}, u_1^{(n-1,L)}) \leq \mathcal{J}(u^{(n-1)}). \end{aligned} \tag{3.28}$$

An alternating minimization

A simple calculation shows that $\mathcal{J}^s_{i,\alpha_i}$ can be written in the following form:

$$\begin{aligned} \mathcal{J}^s_{i,\alpha_i}(u_1 + u_2, a) &= \|T(u_1 + u_2) - g\|_2^2 + 2\alpha|\nabla(u_1 + u_2)|(\Omega) + \alpha_i\|u_i - a\|_2^2 \\ &\quad - \|T(u_i - a)\|_2^2 \\ &= \alpha_i\|u_i - z_i\|_2^2 + 2\alpha|\nabla(u_1 + u_2)|(\Omega) + \phi(a, g, u_{\hat{i}}), \end{aligned}$$

where

$$z_i = \pi_{V_i} a + \frac{1}{\alpha_i}\pi_{V_i}(T^*(g - T(u_{\hat{i}} + a)))$$

and ϕ is a function depending only on $a, g, u_{\hat{i}}$, and $\hat{i} \in \{1, 2\} \setminus \{i\}$. Hence,

$$\arg\min_{u_1 \in V_1} \mathcal{J}^s_{1,\alpha_1}(u_1 + u_2, a) = \arg\min_{u_1 \in V_1} \|u_1 - z_1\|_2^2 + 2\beta_1|\nabla(u_1 + u_2)|(\Omega) \tag{3.29}$$

$$\arg\min_{u_2 \in V_2} \mathcal{J}^s_{2,\alpha_2}(u_1 + u_2, a) = \arg\min_{u_2 \in V_2} \|u_2 - z_2\|_2^2 + 2\beta_2|\nabla(u_1 + u_2)|(\Omega) \tag{3.30}$$

where $\beta_i = \alpha/\alpha_i$, for $i = 1, 2$.

As suggested in the previous section here we solve again the subminimization problems (3.29) and (3.30) by the iterative oblique thresholding algorithm, cf. Theorem 3.1.4, which is specified below in our new setting. We focus, for instance, on the minimization on V_1, and similar statements hold symmetrically for the minimization on V_2.

Theorem 3.2.1 (Oblique thresholding). *For $u_2 \in V_2$ and for $z_1 \in V_1$ the following statements are equivalent:*

(i) $u_1^* = \arg\min_{u_1 \in V_1} \|u_1 - z_1\|_2^2 + 2\beta_1|\nabla(u_1 + u_2)|(\Omega),$

(ii) there exists $\eta_1 \in Range(\pi_{V_2})^* \simeq V_2$ such that $0 \in u_1^* - (z - \eta_1) + \beta_1\partial|\nabla(u_1^* + u_2)|(\Omega)$,

(iii) there exists $\eta_1 \in V_2$ such that $u_1^* = (I - P_{\beta_1 K})(z + u_2 - \eta_1) - u_2 \in V_1$,

(iv) there exists $\eta_1 \in V_2$ such that $\eta_1 = \pi_{V_2} P_{\beta_1 K}(\eta_1 - (z + u_2))$.

The proof of this statement is analogue to the one of Theorem 3.1.4. Note that now $\mathcal{H}$ is a finite dimensional space and thus all the spaces are independent of the particular attached norm and coincide with their duals. Hence all the statements $(i) - (iv)$ are equivalent, cf. Remark 3.1.5.

With this implementation the sequential algorithm in (3.13) produces for a finite number $L, M \in \mathbb{N}$ of inner iterations for each subspace respectively sequences $(u_1^{(n,L)})_n$, $(u_2^{(n,M)})_n$ and $(z_1^{(n,L)})_n$, $(z_2^{(n,M)})_n$ such that

$$z_1^{(n,L)} = u_1^{(n,L-1)} + \frac{1}{\alpha_1}\pi_{V_1}(T^*(g - T(u_1^{(n,L-1)} + u_2^{(n-1,M)}))) \tag{3.31}$$

$$z_2^{(n,M)} = u_2^{(n,M-1)} + \frac{1}{\alpha_2}\pi_{V_2}(T^*(g - T(u_1^{(n,L)} + u_2^{(n,M-1)}))). \tag{3.32}$$

Note that

$$u_1^{(n+1,L)} = \arg\min_{u \in V_1} \|u - z_1^{(n+1,L)}\|_2^2 + 2\beta_1 |\nabla(u + u_2^n)|(\Omega)$$

and

$$u_2^{(n+1,M)} = \arg\min_{u \in V_2} \|u - z_2^{(n+1,M)}\|_2^2 + 2\beta_2 |\nabla(u_1^{(n+1,L)} + u)|(\Omega).$$

3.2.4 Main Result

We do not pursue the analysis of the convergence of the algorithm in (3.13) in our discrete and preconditioned setting, as its proof follows the lines of the one of Theorem 3.1.12. We would like to investigate instead further equivalent conditions for the limits of the sequences produced by this algorithm to be minimizers of $\mathcal{J}$. In order to do that, we need a characterization of solutions of the minimization problem

$$\arg\min_{u \in \mathcal{H}} \mathcal{J}(u),$$

where $\mathcal{J}$ is given as in (3.24). Such a characterization is provided in [113, Proposition 4.1] for the continuous setting. We specify these arguments for our discrete setting and we highlight the significant differences with respect to the continuous one.

Characterization of Solutions

We make the following assumptions:

(A_φ) $\varphi : \mathbb{R} \to \mathbb{R}$ is a convex function, nondecreasing in $\mathbb{R}^+$ such that

(i) $\varphi(0) = 0$.

(ii) There exist $c > 0$ and $b \geq 0$ such that $cz - b \leq \varphi(z) \leq cz + b$, for all $z \in \mathbb{R}^+$.

The particular example we have in mind is simply $\varphi(s) = s$, but we keep a more general notation for uniformity with respect to the continuous version in [113, Proposition 4.1]. In this section we are concerned with the following more general minimization problem

$$\arg\min_{u \in \mathcal{H}} \{\mathcal{J}_\varphi(u) := \|Tu - g\|_2^2 + 2\alpha\varphi(|\nabla u|)(\Omega)\} \tag{3.33}$$

where $g \in \mathcal{H}$ is a datum and $\alpha > 0$ is a fixed constant (in particular for $\varphi(s) = s$).

Proposition 3.2.2. *Let $\zeta, u \in \mathcal{H}$. If the assumption (A_φ) is fulfilled, then $\zeta \in \partial\mathcal{J}_\varphi(u)$ if and only if there exists $M = (M_0, \bar{M}) \in \mathcal{H} \times \mathcal{H}^d$, $\frac{|\bar{M}(x)|}{2\alpha} \leq c_1 \in [0, +\infty)$ for all $x \in \Omega$ such that*

$$\langle \bar{M}(x), (\nabla u)(x)\rangle_{\mathbb{R}^d} + 2\alpha\varphi(|(\nabla u)(x)|) + 2\alpha\varphi_1^*\left(\frac{|\bar{M}(x)|}{2\alpha}\right) = 0 \quad \textit{for all } x \in \Omega \tag{3.34}$$

$$T^* M_0 - \operatorname{div} \bar{M} + \zeta = 0 \tag{3.35}$$

$$-M_0 = 2(Tu - g), \tag{3.36}$$

where φ_1^* is the conjugate function of φ_1 defined by $\varphi_1(s) = \varphi(|s|)$, for $s \in \mathbb{R}$.

If additionally φ is differentiable and $|(\nabla u)(x)| \neq 0$ for $x \in \Omega$, then we can compute $\bar{M}$ as

$$\bar{M}(x) = -2\alpha \frac{\varphi'(|(\nabla u)(x)|)}{|(\nabla u)(x)|}(\nabla u)(x). \tag{3.37}$$

Proof. It is clear that $\zeta \in \partial\mathcal{J}_\varphi(u)$ if and only if $u = \arg\min_{v\in\mathcal{H}}\{\mathcal{J}_\varphi(v) - \langle\zeta, v\rangle_{\mathcal{H}}\}$, and let us consider the following variational problem:

$$\inf_{v \in \mathcal{H}}\{\mathcal{J}_\varphi(v) - \langle\zeta, v\rangle_{\mathcal{H}}\} = \inf_{v\in\mathcal{H}}\{\|Tv - g\|_2^2 + 2\alpha\varphi(|\nabla v|)(\Omega) - \langle\zeta, v\rangle_{\mathcal{H}}\}. \tag{$\mathbf{P}$}$$

We denote such an infimum by $\inf(\mathbf{P})$. Now we compute $(\mathbf{P}^*)$ the dual of $(\mathbf{P})$. Let $\mathcal{F} : \mathcal{H} \to \mathbb{R}$, $\mathcal{G} : \mathcal{H} \times \mathcal{H}^d \to \mathbb{R}$, $\mathcal{G}_1 : \mathcal{H} \to \mathbb{R}$, $\mathcal{G}_2 : \mathcal{H}^d \to \mathbb{R}$, such that

$$\begin{aligned}
\mathcal{F}(v) &= -\langle\zeta, v\rangle_{\mathcal{H}}\\
\mathcal{G}_1(w_0) &= \|w_0 - g\|_2^2\\
\mathcal{G}_2(\bar{w}) &= 2\alpha\varphi(|\bar{w}|)(\Omega)\\
\mathcal{G}(w) &= \mathcal{G}_1(w_0) + \mathcal{G}_2(\bar{w})
\end{aligned}$$

with $w = (w_0, \bar{w}) \in \mathcal{H} \times \mathcal{H}^d$. Then the dual problem of $(\mathbf{P})$ is given by (cf. Section 2.2.3)

$$\sup_{p^* \in \mathcal{H}\times\mathcal{H}^d} \{-\mathcal{F}^*(\Lambda^* p^*) - \mathcal{G}^*(-p^*)\}, \tag{$\mathbf{P}^*$}$$

where $\Lambda : \mathcal{H} \to \mathcal{H} \times \mathcal{H}^d$ is defined by

$$\Lambda v = (Tv, (\nabla v)^1, \ldots, (\nabla v)^d)$$

and Λ^* is its adjoint. We denote the supremum in ($\mathbf{P}^*$) by sup($\mathbf{P}^*$). Using the definition of the conjugate function we compute $\mathcal{F}^*$ and $\mathcal{G}^*$. In particular

$$\mathcal{F}^*(\Lambda^* p^*) = \sup_{v\in\mathcal{H}}\{\langle \Lambda^* p^*, v\rangle_{\mathcal{H}} - \mathcal{F}(v)\} = \sup_{v\in\mathcal{H}}\langle \Lambda^* p^* + \zeta, v\rangle_{\mathcal{H}} = \begin{cases} 0 & \text{if } \Lambda^* p^* + \zeta = 0 \\ \infty & \text{otherwise} \end{cases},$$

where $p^* = (p_0^*, \bar{p}^*)$ and

$$\begin{aligned}
\mathcal{G}^*(p^*) &= \sup_{w\in\mathcal{H}\times\mathcal{H}^d}\{\langle p^*, w\rangle_{\mathcal{H}\times\mathcal{H}^d} - \mathcal{G}(w)\} \\
&= \sup_{w=(w_0,\bar{w})\in\mathcal{H}\times\mathcal{H}^d}\{\langle p_0^*, w_0\rangle_{\mathcal{H}} + \langle \bar{p}^*, \bar{w}\rangle_{\mathcal{H}^d} - \mathcal{G}_1(w_0) - \mathcal{G}_2(\bar{w})\} \\
&= \sup_{w_0\in\mathcal{H}}\{\langle p_0^*, w_0\rangle_{\mathcal{H}} - \mathcal{G}_1(w_0)\} + \sup_{\bar{w}\in\mathcal{H}^d}\{\langle \bar{p}^*, \bar{w}\rangle_{\mathcal{H}^d} - \mathcal{G}_2(\bar{w})\} \\
&= \mathcal{G}_1^*(p_0^*) + \mathcal{G}_2^*(\bar{p}^*).
\end{aligned}$$

We have that

$$\mathcal{G}_1^*(p_0^*) = \left\langle \frac{p_0^*}{4} + g, p_0^* \right\rangle_{\mathcal{H}}$$

and

$$\mathcal{G}_2^*(\bar{p}^*) = 2\alpha\varphi_1^*\left(\frac{|\bar{p}^*|}{2\alpha}\right)(\Omega)$$

if $\frac{|\bar{p}^*(x)|}{2\alpha} \in \operatorname{Dom}\varphi_1^*$, where φ_1^* is the conjugate function of φ_1 defined by

$$\varphi_1(s) := \varphi(|s|) \quad s \in \mathbb{R}.$$

See Section 2.2.2 for the explicit computation of these conjugate functions. So we can write ($\mathbf{P}^*$) in the following way

$$\sup_{p^*\in\mathcal{K}}\left\{-\left\langle \frac{-p_0^*}{4} + g, -p_0^* \right\rangle_{\mathcal{H}} - 2\alpha\varphi_1^*\left(\frac{|\bar{p}^*|}{2\alpha}\right)(\Omega)\right\}, \tag{3.38}$$

where

$$\mathcal{K} = \left\{p^* \in \mathcal{H}\times\mathcal{H}^d : \frac{|\bar{p}^*(x)|}{2\alpha} \in \operatorname{Dom}\varphi_1^* \text{ for all } x\in\Omega, \Lambda^* p^* + \zeta = 0\right\}.$$

The function φ_1 also fulfills assumption (A_φ)(ii) (i.e., there exists $c_1 > 0, b \geq 0$ such that $c_1 z - b \leq \varphi_1(z) \leq c_1 z + b$, for all $z \in \mathbb{R}^+$). The conjugate function of φ_1 is given by $\varphi_1^*(s) = \sup_{z\in\mathbb{R}}\{\langle s, z\rangle - \varphi_1(z)\}$. By using the previous inequalities and by noting that φ_1 is even (i.e., $\varphi_1(z) = \varphi_1(-z)$ for all $z \in \mathbb{R}$) we have

$$\begin{aligned}
(\sup_{z\in\mathbb{R}}\{\langle s, z\rangle - c_1|z| + b\} \geq) \sup_{z\in\mathbb{R}}\{\langle s, z\rangle - \varphi_1(z)\} &\geq \sup_{z\in\mathbb{R}}\{\langle s, z\rangle - c_1|z| - b\} \\
&= \begin{cases} -b & \text{if } |s| \leq c_1 \\ \infty & \text{otherwise} \end{cases}.
\end{aligned} \tag{3.39}$$

In particular, one can see that $s \in \operatorname{Dom} \varphi_1^*$ if and only if $|s| \le c_1$.

From $\Lambda^* p^* + \zeta = 0$ we obtain

$$\begin{aligned}\langle \Lambda^* p^*, \omega\rangle_{\mathcal{H}} + \langle \zeta, \omega\rangle_{\mathcal{H}} &= \langle p^*, \Lambda\omega\rangle_{\mathcal{H}^{d+1}} + \langle \zeta, \omega\rangle_{\mathcal{H}}\\ &= \langle p_0^*, T\omega\rangle_{\mathcal{H}} + \langle \bar{p}^*, \nabla\omega\rangle_{\mathcal{H}^d} + \langle \zeta, \omega\rangle_{\mathcal{H}}\\ &= 0\end{aligned}$$

for all $\omega \in \mathcal{H}$. Then, since $\langle \bar{p}^*, \nabla\omega\rangle_{\mathcal{H}^d} = \langle -\operatorname{div} \bar{p}^*, \omega\rangle_{\mathcal{H}}$ (see Section 3.2.2), we have

$$T^* p_0^* - \operatorname{div} \bar{p}^* + \zeta = 0.$$

Hence we can write $\mathcal{K}$ in the following way

$$\mathcal{K} = \left\{ p^* = (p_0^*, \bar{p}^*) \in \mathcal{H} \times \mathcal{H}^d : \frac{|\bar{p}^*(x)|}{2\alpha} \le c_1 \text{ for all } x \in \Omega, T^* p_0^* - \operatorname{div} \bar{p}^* + \zeta = 0 \right\}.$$

We now apply the duality results from Theorem 2.2.5, since the functional in (**P**) is convex, continuous with respect to Λv in $\mathcal{H} \times \mathcal{H}^d$, and inf(**P**) is finite. Then inf(**P**)= sup(**P***)$\in \mathbb{R}$ and (**P***) has a solution $M = (M_0, \bar{M}) \in \mathcal{K}$.

Let us assume that u is a solution of (**P**) and M is a solution of (**P***). From inf(**P**)= sup(**P***) we get

$$\|Tu - g\|_2^2 + 2\alpha\varphi(|\nabla u|)(\Omega) - \langle \zeta, u\rangle_{\mathcal{H}} = -\left\langle \frac{-M_0}{4} + g, -M_0 \right\rangle_{\mathcal{H}} - 2\alpha\varphi_1^*\left(\frac{|\bar{M}|}{2\alpha}\right)(\Omega), \tag{3.40}$$

where $M = (M_0, \bar{M}) \in \mathcal{H} \times \mathcal{H}^d$, $\frac{|\bar{M}(x)|}{2\alpha} \le c_1$ and $T^* M_0 - \operatorname{div} \bar{M} + \zeta = 0$, which verifies the direct implication of (3.35). In particular

$$-\langle \zeta, u\rangle_{\mathcal{H}} = \langle T^* M_0, u\rangle_{\mathcal{H}} - \langle \operatorname{div} \bar{M}, u\rangle_{\mathcal{H}} = \langle M_0, Tu\rangle_{\mathcal{H}} + \langle \bar{M}, \nabla u\rangle_{\mathcal{H}^d},$$

and

$$\begin{aligned}&\|Tu - g\|_2^2 + \langle M_0, Tu\rangle_{\mathcal{H}} + \langle \bar{M}, \nabla u\rangle_{\mathcal{H}^d} + 2\alpha\varphi(|\nabla u|)(\Omega)\\ &\qquad + \left\langle \frac{-M_0}{4} + g, -M_0 \right\rangle_{\mathcal{H}} + 2\alpha\varphi_1^*\left(\frac{|\bar{M}|}{2\alpha}\right)(\Omega) = 0.\end{aligned} \tag{3.41}$$

Let us write (3.41) again in the following form

$$\begin{aligned}&\sum_{x\in\Omega} |(Tu - g)(x)|^2 + \sum_{x\in\Omega} M_0(x)(Tu)(x) + \sum_{x\in\Omega}\sum_{j=1}^{d} \bar{M}^j(x)(\nabla u)^j(x)\\ &\qquad + \sum_{x\in\Omega} 2\alpha\varphi(|(\nabla u)(x)|) + \sum_{x\in\Omega}\left(\frac{-M_0(x)}{4} + g(x)\right)(-M_0(x))\\ &\qquad\qquad + \sum_{x\in\Omega} 2\alpha\varphi_1^*\left(\frac{|\bar{M}(x)|}{2\alpha}\right) = 0.\end{aligned} \tag{3.42}$$

Now we have

1. $2\alpha\varphi(|(\nabla u)(x)|) + \sum_{j=1}^{d} \bar{M}^j(x)(\nabla u)^j(x) + 2\alpha\varphi_1^*\left(\frac{|\bar{M}(x)|}{2\alpha}\right) \geq 2\alpha\varphi(|(\nabla u)(x)|) - \sum_{j=1}^{d} |\bar{M}^j(x)||(\nabla u)^j(x)| + 2\alpha\varphi_1^*\left(\frac{|\bar{M}(x)|}{2\alpha}\right) \geq 0$ by the definition of φ_1^*, since
$$\begin{aligned} 2\alpha\varphi_1^*\left(\frac{|\bar{M}(x)|}{2\alpha}\right) &= \sup_{S\in\mathbb{R}^d}\{\langle \bar{M}(x), S\rangle_{\mathbb{R}^d} - 2\alpha\varphi(|S|)\} \\ &= \sup_{S=(S^1,\ldots,S^d)\in\mathbb{R}^d} \{\sum_{j=1}^{d} |\bar{M}^j(x)||S^j| - 2\alpha\varphi(|S|)\} \end{aligned}$$
2. $|(Tu-g)(x)|^2 + M_0(x)(Tu)(x) + \left(\frac{-M_0(x)}{4} + g(x))(-M_0(x)\right) = (((Tu)(x) - g(x)))^2 + M_0(x)((Tu)(x)-g(x)) + \left(\frac{M_0(x)}{2}\right)^2 = \left(((Tu)(x)-g(x)) + \frac{M_0(x)}{2}\right)^2 \geq 0.$

Hence condition (3.41) reduces to

$$2\alpha\varphi(|(\nabla u)(x)|) + \sum_{j=1}^{d} \bar{M}^j(x)(\nabla u)^j(x) + 2\alpha\varphi_1^*\left(\frac{|\bar{M}(x)|}{2\alpha}\right) = 0 \quad \text{for all } x \in \Omega \tag{3.43}$$

$$-M_0(x) = 2((Tu)(x) - g(x)) \quad \text{for all } x \in \Omega. \tag{3.44}$$

Conversely, if such an $M = (M_0, \bar{M}) \in \mathcal{H} \times \mathcal{H}^d$ with $\frac{|\bar{M}(x)|}{2\alpha} \leq c_1$ that fulfills conditions (3.34)-(3.36) exists, it is clear from previous considerations that equation (3.40) holds. Let us denote the functional on the left side of (3.40) by

$$P(u) := \|Tu - g\|_2^2 + 2\alpha\varphi(|\nabla u|)(\Omega) - \langle \zeta, u\rangle_{\mathcal{H}}$$

and the functional on the right side of (3.40) by

$$P^*(M) := -\left\langle \frac{-M_0}{4} + g, -M_0 \right\rangle_{\mathcal{H}} - 2\alpha\varphi_1^*\left(\frac{|\bar{M}|}{2\alpha}\right)(\Omega).$$

We know that the functional P is the functional of $(\mathbf{P})$ and P^* is the functional of $(\mathbf{P}^*)$. Hence $\inf P = \inf(\mathbf{P})$ and $\sup P^* = \sup(\mathbf{P}^*)$. Since P is convex, continuous with respect to Λu in $\mathcal{H} \times \mathcal{H}^d$, and $\inf(\mathbf{P})$ is finite, we know from Theorem 2.2.5 that $\inf(\mathbf{P}) = \sup(\mathbf{P}^*) \in \mathbb{R}$. We assume that M is no solution of $(\mathbf{P}^*)$, i.e., $P^*(M) < \sup(\mathbf{P}^*)$, and u is no solution of $(\mathbf{P})$, i.e, $P(u) > \inf(\mathbf{P})$. Then we have that

$$P(u) > \inf{(\mathbf{P})} = \sup{(\mathbf{P}^*)} > P^*(M).$$

Thus (3.40) is valid if and only if M is a solution of $(\mathbf{P}^*)$ and u is a solution of $(\mathbf{P})$, which amounts to saying that $\zeta \in \partial\mathcal{J}_\varphi(u)$.

If additionally φ is differentiable and $|(\nabla u)(x)| \neq 0$ for $x \in \Omega$, we show that we can compute $\bar{M}(x)$ explicitly. From equation (3.34) (resp. (3.43)) we have

$$2\alpha\varphi_1^*\left(\frac{|-\bar{M}(x)|}{2\alpha}\right) = -\langle \bar{M}(x), (\nabla u)(x)\rangle_{\mathbb{R}^d} - 2\alpha\varphi(|(\nabla u)(x)|). \tag{3.45}$$

From the definition of conjugate function we have

$$\begin{aligned}
2\alpha\varphi_1^*\left(\frac{|-\bar{M}(x)|}{2\alpha}\right) &= 2\alpha \sup_{t\in\mathbb{R}} \left\{\left\langle \frac{|-\bar{M}(x)|}{2\alpha}, t\right\rangle - \varphi_1(t)\right\} \\
&= 2\alpha \sup_{t\geq 0} \left\{\left\langle \frac{|-\bar{M}(x)|}{2\alpha}, t\right\rangle - \varphi_1(t)\right\} \\
&= 2\alpha \sup_{t\geq 0} \sup_{\substack{S\in\mathbb{R}^d \\ |S|=t}} \left\{\left\langle \frac{-\bar{M}(x)}{2\alpha}, S\right\rangle_{\mathbb{R}^d} - \varphi_1(|S|)\right\} \\
&= \sup_{S\in\mathbb{R}^d} \left\{\left\langle -\bar{M}(x), S\right\rangle_{\mathbb{R}^d} - 2\alpha\varphi(|S|)\right\}.
\end{aligned} \tag{3.46}$$

Now, if $|(\nabla u)(x)| \neq 0$ for $x \in \Omega$, then it follows from (3.45) that the supremum is taken in $S = |(\nabla u)(x)|$ and we have

$$\nabla_S(-\langle \bar{M}(x), S\rangle_{\mathbb{R}^d} - 2\alpha\varphi(|S|)) = 0,$$

which implies

$$\bar{M}^j(x) = -2\alpha\frac{\varphi'(|(\nabla u)(x)|)}{|(\nabla u)(x)|}(\nabla u)^j(x) \quad j = 1, \dots, d,$$

and verifies (3.37). This finishes the proof. □

Remark 3.2.3. *(i) For $\varphi(s) = s$, the function φ_1 from Proposition 3.2.2 turns out to be $\varphi_1(s) = |s|$. Its conjugate function φ_1^* is then given by*

$$\varphi_1^*(s^*) = \sup_{s\in\mathbb{R}}\{\langle s^*, s\rangle - |s|\} = \begin{cases} 0 & \text{for } |s^*| \leq 1 \\ \infty & \text{else} \end{cases}.$$

Hence condition (3.34) *specifies as follows*

$$\langle \bar{M}(x), (\nabla u)(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u)(x)| = 0$$

and, directly from the proof of Proposition 3.2.2, $|\bar{M}(x)| \leq 2\alpha$ for all $x \in \Omega$.

(ii) We want to highlight a few important differences with respect to the continuous case. Due to our definition of the gradient and its relationship with the divergence operator $-\operatorname{div} = \nabla^*$ no boundary conditions are needed. Therefore condition (10) of [113, Proposition 4.1] has no discrete correspondent in

our setting. The continuous total variation of a function can be decomposed into an absolute continuous part with respect to the Lebesgue measure and a singular part, whereas no singular part appears in the discrete setting. Therefore condition (6) and (7) of [113, Proposition 4.1] does not have a discrete correspondent neither.

(iii) *An interesting consequence of Proposition 3.2.2 is that the map $\mathbb{S}_\alpha = (I - P_{\alpha K})$ is bounded, i.e., $\|\mathbb{S}_\alpha(z^k)\|_2 \to \infty$ if and only if $\|z^k\|_2 \to \infty$, for $k \to \infty$. In fact, since*

$$\mathbb{S}_\alpha(z) \quad = \quad \arg\min_{u \in \mathcal{H}} \|u - z\|_2^2 + 2\alpha |\nabla u|(\Omega),$$

from (3.35) *and* (3.36), *we immediately obtain*

$$\mathbb{S}_\alpha(z) = z - \frac{1}{2} \operatorname{div} \bar{M},$$

and thus $\bar{M}$ is uniformly bounded.

Theorem 3.2.4. *Let $(u^{(n)})_n$ be a sequence produced by* (3.13). *Then for a strongly convergent subsequence of $(u^{(n)} = u_1^{(n,L)} + u_2^{(n,M)})_n$ with limit $u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)}$, we have*

$$u_1^{(\infty)} \quad = \quad \arg\min_{u \in V_1} \|u - z_1^{(\infty)}\|_2^2 + 2\beta_1 |\nabla(u + u_2^{(\infty)})|(\Omega), \tag{3.47}$$

$$u_2^{(\infty)} \quad = \quad \arg\min_{u \in V_2} \|u - z_2^{(\infty)}\|_2^2 + 2\beta_2 |\nabla(u_1^{(\infty)} + u)|(\Omega), \tag{3.48}$$

$$z_1^{(\infty)} \quad = \quad u_1^{(\infty)} + \frac{1}{\alpha_1} \pi_{V_1}(T^*(g - Tu^{(\infty)})), \tag{3.49}$$

$$z_2^{(\infty)} \quad = \quad u_2^{(\infty)} + \frac{1}{\alpha_2} \pi_{V_2}(T^*(g - Tu^{(\infty)})), \tag{3.50}$$

where $\beta_i = \alpha/\alpha_i$, for $i = 1, 2$.

Moreover let us denote $z^{(\infty)} = u^{(\infty)} + T^*(g - Tu^{(\infty)})$. Then, $u^{(\infty)}$ is a minimizer of (3.24) *if and only if*

$$u^{(\infty)} = \arg\min_{u \in \mathcal{H}} \{\mathcal{F}(u) := \|u - z^{(\infty)}\|_2^2 + 2\alpha |\nabla u|(\Omega)\}. \tag{3.51}$$

Before proving the previous statements we add some comments on the possibility of verification of the minimality condition (3.51). Let $F(u_1, u_2) = \mathcal{F}(u_1 + u_2)$ for $u_1 \in V_1$ and $u_2 \in V_2$. Then (3.47) and (3.48) imply

$$F(u_1^{(\infty)}, u_2^{(\infty)}) \leq \arg \min_{v_1 \in V_1, v_2 \in V_2} \Big\{ F(v_1, u_2^{(\infty)}), F(u_1^{(\infty)}, v_2) \Big\}. \tag{3.52}$$

Unfortunately, (3.52) may not imply that $u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)}$ is a minimizer of (3.51) and eventually of (3.24). We propose the following univariate counterexample, which also shows that the algorithm in (3.13) may fail to converge to a minimizing solution. For simplicity, we return to the continuous setting and we assume that Ω is the interval $[-1, 2]$, and that $g = 1_{[0,1/2)}$. We consider univariate Haar wavelets, i.e., let $\varphi_0 = 1_{[0,1)}$ and $\psi_0 = 1_{[0,1/2)} - 1_{[1/2,1)}$. Then we have

$$g = \frac{1}{2}\varphi_0 + \frac{1}{2}\psi_0.$$

Counterexample

The subspace correction method in (3.2) was introduced in [66] particularly for ℓ_1-minimization and total variation minimization. However it was only possible to show convergence of this algorithm under some technical conditions which are in general not fulfilled, see Theorem 3.1.12. Now in the following proposition we state a very interesting counterexample, which shows that the subspace correction method in (3.2) considered for the case of total variation minimization does not in general converge to an expected minimizer.

Proposition 3.2.5. *Let $0 < \alpha < 1/8$. In addition, let V_1 be the subspace of $L^2([-1,2])$ generated by $\{\varphi_0(x-k) : k \in \{-1,0,1\}\}$ and let V_2 be the subspace of $L^2([-1,2])$ generated by $\{\psi_{j,k}(x) = 2^{j/2}\psi_0(2^jx-k) : j \in \mathbb{Z}^+ \cup \{0\}, k \in \{-2^j, \ldots, 2^j\}\}$, then*

$$u_1^{(\infty)} = \frac{1-4\alpha}{2}\varphi_0, \quad u_2^{(\infty)} = \frac{1-4\alpha}{2}\psi_0,$$

which satisfy

$$\arg\min_{\substack{u_1\in V_1\\u_2\in V_2}} F(u_1,u_2) < F(u_1^{(\infty)}, u_2^{(\infty)}) \le \arg\min_{\substack{v_1\in V_1\\v_2\in V_2}} \left\{F(v_1, u_2^{(\infty)}), F(u_1^{(\infty)}, v_2)\right\}, \quad (3.53)$$

where

$$\begin{aligned} F(u_1,u_2) &= \mathcal{F}(u_1+u_2) = \|u_1+u_2-g\|_2^2 + 2\alpha|\nabla(u_1+u_2)|([-1,2]) \\ &= \left\|u_1 - \frac{1}{2}\varphi_0\right\|_2^2 + \left\|u_2 - \frac{1}{2}\psi_0\right\|_2^2 + 2\alpha|\nabla(u_1+u_2)|([-1,2]). \end{aligned}$$

Proof. We prove the result by showing that the algorithm in (3.13), starting with $u^{(0)} = 0$, stops by converging to $u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)}$ in finite iterations, and that (3.53) holds. Let $u_1^{(0)} = u_2^{(0)} = 0$. Then

$$u_1^{(1)} = \arg\min_{u\in V_1} \left\|u - \frac{1}{2}\varphi_0\right\|_2^2 + 2\alpha|\nabla u|([-1,2]). \quad (3.54)$$

Then $u_1^{(1)} = a\varphi_0$ for some $a > 0$ and

$$\begin{aligned} \left\|u - \frac{1}{2}\varphi_0\right\|_2^2 + 2\alpha|\nabla u|([-1,2]) &= \left\|a\varphi_0 - \frac{1}{2}\varphi_0\right\|_2^2 + 2\alpha a|\nabla\varphi_0|([-1,2]) \\ &= \left(a - \frac{1}{2}\right)^2 + 4\alpha a = \left(a + \frac{4\alpha - 1}{2}\right)^2 + 2\alpha - 4\alpha^2. \end{aligned}$$

Since $\alpha < 1/8$, (3.54) attains its minimum when

$$a = \frac{1-4\alpha}{2}, \text{ i.e., } u_1^{(1)} = \frac{1-4\alpha}{2}\varphi_0.$$

Now, we solve

$$u_2^{(1)} = \arg\min_{u\in V_2} \left\| u - \frac{1}{2}\psi_0 \right\|_2^2 + 2\alpha |\nabla(u_1^{(1)} + u)|([-1,2]). \tag{3.55}$$

It is not hard to see that $u_2^{(1)} = b\psi_0$ for some $b > 0$. If we assume $b \le \frac{1-4\alpha}{2}$, then

$$\begin{aligned} &\left\| u - \frac{1}{2}\psi_0 \right\|_2^2 + 2\alpha |\nabla(u_1^{(1)} + u)|([-1,2]) \\ &= \left(b - \frac{1}{2}\right)^2 + 2\alpha\left(\frac{1-4\alpha}{2} + b + 2b + \frac{1-4\alpha}{2} - b\right) \\ &= \left(b + \frac{4\alpha - 1}{2}\right)^2 + 4\alpha - 12\alpha^2 \ge 4\alpha - 12\alpha^2, \end{aligned}$$

which is minimized when $b = \frac{1-4\alpha}{2}$. On the other hand, if we assume $b \ge \frac{1-4\alpha}{2}$, then since $0 < \frac{1-8\alpha}{2} < \frac{1-4\alpha}{2} \le b$,

$$\begin{aligned} &\left\| u - \frac{1}{2}\psi_0 \right\|_2^2 + 2\alpha |\nabla(u_1^{(1)} + u)|([-1,2]) \\ &= \left(b - \frac{1}{2}\right)^2 + 2\alpha\left(\frac{1-4\alpha}{2} + b + 2b - \frac{1-4\alpha}{2} + b\right) \\ &= \left(b + \frac{8\alpha - 1}{2}\right)^2 + 4\alpha - 16\alpha^2 \ge 4\alpha - 12\alpha^2, \end{aligned}$$

which is also minimized when $b = \frac{1-4\alpha}{2}$. Hence

$$u_2^{(1)} = \frac{1-4\alpha}{2}\psi_0.$$

Now, we solve

$$u_1^{(2)} = \arg\min_{u\in V_1} \left\| u - \frac{1}{2}\varphi_0 \right\|_2^2 + 2\alpha |\nabla(u + u_2^{(1)})|([-1,2]).$$

It is easy to see that $u_1^{(2)} = a\varphi_0$ for some $a > 0$. If we assume $a \le \frac{1-4\alpha}{2}$, then since $\frac{1-4\alpha}{2} \le \frac{1}{2}$,

$$\begin{aligned} &\left\| u - \frac{1}{2}\varphi_0 \right\|_2^2 + 2\alpha |\nabla(u + u_2^{(1)})|([-1,2]) \\ &= \left(a - \frac{1}{2}\right)^2 + 2\alpha\left(a + \frac{1-4\alpha}{2} + (1-4\alpha) + \frac{1-4\alpha}{2} - a\right) \\ &= \left(a - \frac{1}{2}\right)^2 + 4\alpha(1-4\alpha) \ge 4\alpha - 12\alpha^2, \end{aligned}$$

which is minimized when $a = \frac{1-4\alpha}{2}$. On the other hand, if we assume $a \geq \frac{1-4\alpha}{2}$, then

$$\begin{aligned}
&\left\| u - \frac{1}{2}\varphi_0 \right\|_2^2 + 2\alpha|\nabla(u + u_2^{(1)})|([-1,2]) \\
&= \left(a - \frac{1}{2}\right)^2 + 2\alpha\left(a + \frac{1-4\alpha}{2} + (1-4\alpha) + a - \frac{1-4\alpha}{2}\right) \\
&= \left(a + \frac{4\alpha - 1}{2}\right)^2 + 4\alpha - 12\alpha^2 \geq 4\alpha - 12\alpha^2,
\end{aligned}$$

which is also minimized when $a = \frac{1-4\alpha}{2}$. We finally obtain

$$u_1^{(2)} = \frac{1-4\alpha}{2}\varphi_0 = u_1^{(1)}.$$

Therefore, after only one step of the algorithm in (3.13), we have

$$u_1^{(\infty)} = \frac{1-4\alpha}{2}\varphi_0, \quad u_2^{(\infty)} = \frac{1-4\alpha}{2}\psi_0.$$

It is now easy to see that $u_1^{(\infty)}, u_2^{(\infty)}$ satisfy (3.52) and

$$F(u_1^{(\infty)}, u_2^{(\infty)}) = 4\alpha - 8\alpha^2.$$

However, if $u = a1_{[0,1/2)} = \frac{a}{2}\varphi_0 + \frac{a}{2}\psi_0$, then

$$\begin{aligned}
\mathcal{F}(u) &= \|u - g\|_2^2 + 2\alpha|\nabla u|([-1,2]) = (a-1)^2\|1_{[0,1/2)}\|_2^2 + 2\alpha \cdot 2a \\
&= \frac{1}{4}(a-1)^2 + 4\alpha a = \frac{1}{4}(a + (8\alpha - 1))^2 + 4\alpha - 16\alpha^2.
\end{aligned}$$

Since $0 < \alpha < 1/8$, if we set $u_0 = (1-8\alpha)1_{[0,1/2)} = \frac{1-8\alpha}{2}\varphi_0 + \frac{1-8\alpha}{2}\psi_0$, then

$$\begin{aligned}
\min_{u_1 \in V_1, u_2 \in V_2} F(u_1, u_2) \leq \mathcal{F}(u_0) = 4\alpha - 16\alpha^2 < 4\alpha - 8\alpha^2 = \mathcal{F}(u_1^{(\infty)} + u_2^{(\infty)}) \\
= F(u_1^{(\infty)}, u_2^{(\infty)}).
\end{aligned}$$

□

Proposition 3.2.2 also provides us with the following useful characterization.

Corollary 3.2.6. *The subdifferential of $\alpha\partial|\nabla u|(\Omega)$ is fully characterized by*

$$\begin{aligned}
\alpha\partial|\nabla u|(\Omega) &= \{\operatorname{div}(\xi) \in \mathcal{H} : \|\xi\|_\infty \leq \alpha,\ \langle \xi(x), \nabla u(x)\rangle_{\mathbb{R}^d} + \alpha|\nabla u|(x) = 0 \textit{ for all } x \in \Omega\} \\
&= \{\operatorname{div}(\xi) \in \mathcal{H} : -\operatorname{div}(\xi) = P_{\alpha K}(-u - \operatorname{div}(\xi))\}.
\end{aligned}$$

Proof. If we consider $T = I$ and $\phi(s) = s$ in Proposition 3.2.2, then $\tilde{\zeta} \in \alpha\partial|\nabla u|(\Omega)$ if and only if $\zeta = 2(\tilde{\zeta} + u - g) \in \partial\mathcal{J}(u)$ if and only if there exists $(\xi_0, \xi) \in \mathcal{H} \times \mathcal{H}^d$ such that

1. $\|\xi\|_\infty \leq \alpha$,
2. $\langle \xi(x), \nabla u(x)\rangle_{\mathbb{R}^d} + \alpha|\nabla u(x)| = 0$ for all $x \in \Omega$,
3. $\tilde{\zeta} = \operatorname{div}(\xi)$.

Hence,

$$\alpha\partial|\nabla u|(\Omega) = \{\operatorname{div}(\xi) \in \mathcal{H} : \|\xi\|_\infty \leq \alpha,\ \langle \xi(x), \nabla u(x)\rangle_{\mathbb{R}^d} + \alpha|\nabla u|(x) = 0 \text{ for all } x \in \Omega\}.$$

We also notice that

$$\operatorname{div}(\xi) \in \alpha|\nabla u|(\Omega) \text{ if and only if } 0 \in u - (u + \operatorname{div}(\xi)) + \alpha\partial|\nabla u|(\Omega),$$

which is equivalent to

$$u = \arg\min_v \|v - (u + \operatorname{div}(\xi))\|_2^2 + 2\alpha|\nabla v|(\Omega),$$

that is,

$$-u = \arg\min_v \|v + (u + \operatorname{div}(\xi))\|_2^2 + 2\alpha|\nabla v|(\Omega).$$

By Remark 3.1.2, the latter optimality problem is equivalent to

$$-u = (I - P_{\alpha K})(-u - \operatorname{div}(\xi)),$$

that is,

$$-\operatorname{div}(\xi) = P_{\alpha K}(-u - \operatorname{div}(\xi)).$$

Therefore, we also have

$$\alpha\partial|\nabla u|(\Omega) = \{\operatorname{div}(\xi) \in \mathcal{H} : -\operatorname{div}(\xi) = P_{\alpha K}(-u - \operatorname{div}(\xi))\}.$$

□

3.2.5 Proof of Theorem 3.2.4

For simplicity, we rename a convergent subsequence again by $(u^{(n)} = u_1^{(n,L)} + u_2^{(n,M)})_n$. Equations (3.49) and (3.50) follow directly from (3.31) and (3.32) for $n \to \infty$ by using (3.16), which states the asymptotic regularity of the sequence, i.e.,

$$\left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_2^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_2^2\right) \to 0, \quad n \to \infty.$$

Furthermore, it is also easy to see that for any $u_1 \in V_1$,

$$\begin{aligned}\|u_1^{(\infty)} - z_1^{(\infty)}\|_2^2 + 2\beta_1|\nabla u^{(\infty)}|(\Omega) &= \lim_{n\to\infty} \|u_1^{(n+1,L)} - z_1^{(n+1,L)}\|_2^2 \\ &\quad + 2\beta_1|\nabla(u_1^{(n+1,L)} + u_2^{(n,M)})|(\Omega) \\ &\leq \lim_{n\to\infty} \|u_1 - z_1^{(n+1,L)}\|_2^2 + 2\beta_1|\nabla(u_1 + u_2^{(n,M)})|(\Omega) \\ &= \|u_1 - z_1^{(\infty)}\|_2^2 + 2\beta_1|\nabla(u_1 + u_2^{(\infty)})|(\Omega).\end{aligned}$$

Hence, we have

$$u_1^{(\infty)} = \arg\min_{u \in V_1} \|u - z_1^{(\infty)}\|_2^2 + 2\beta_1 |\nabla(u + u_2^{(\infty)})|(\Omega).$$

With the same argument one obtains (3.48). By Theorem 3.1.4 the optimality conditions (3.47) and (3.48) are equivalent to

$$\begin{aligned} 0 &\in u_1^{(\infty)} - (z_1^{(\infty)} - \eta_1^{(\infty)}) + \beta_1 \partial |\nabla u^{(\infty)}|(\Omega), \\ 0 &\in u_2^{(\infty)} - (z_2^{(\infty)} - \eta_2^{(\infty)}) + \beta_2 \partial |\nabla u^{(\infty)}|(\Omega). \end{aligned}$$

Then by Corollary 3.2.6 there exist ξ_1, ξ_2 such that

$$\begin{aligned} \operatorname{div}(\xi_1) &= -u_1^{(\infty)} + (z_1^{(\infty)} - \eta_1^{(\infty)}), && (3.56) \\ \operatorname{div}(\xi_2) &= -u_2^{(\infty)} + (z_2^{(\infty)} - \eta_2^{(\infty)}), && (3.57) \end{aligned}$$

and with the following additional properties

1. $\|\xi_1\|_\infty \leq \beta_1$, $\|\xi_2\|_\infty \leq \beta_2$ and
2. $\langle \xi_i(x), \nabla u^{(\infty)}(x) \rangle_{\mathbb{R}^d} + \beta_i |\nabla u^{(\infty)}(x)| = 0$ for all $x \in \Omega$ and $i = 1, 2$.

Multiplying (3.56) by α_1 and (3.57) by α_2 yields

$$\begin{aligned} -\alpha_1 u_1^{(\infty)} + \alpha_1 z_1^{(\infty)} - \alpha_1 \eta_1^{(\infty)} - \alpha_1 \operatorname{div}(\xi_1) &= 0, \\ -\alpha_2 u_2^{(\infty)} + \alpha_2 z_2^{(\infty)} - \alpha_2 \eta_2^{(\infty)} - \alpha_2 \operatorname{div}(\xi_2) &= 0. \end{aligned}$$

If we sum up the last two equations we obtain

$$-\alpha_1 u_1^{(\infty)} + \alpha_1 z_1^{(\infty)} - \alpha_2 u_2^{(\infty)} + \alpha_2 z_2^{(\infty)} - \operatorname{div}(\alpha_1 \xi_1) - \operatorname{div}(\alpha_2 \xi_2) - (\alpha_1 \eta_1^{(\infty)} + \alpha_2 \eta_2^{(\infty)}) = 0. \tag{3.58}$$

From Theorem 3.1.4 we have that

$$\eta_1^{(\infty)} = \pi_{V_2} P_{\beta_1 K}(\eta_1^{(\infty)} - (z_1^{(\infty)} + u_2^{(\infty)})) \text{ and } \eta_2^{(\infty)} = \pi_{V_1} P_{\beta_2 K}(\eta_2^{(\infty)} - (z_2^{(\infty)} + u_1^{(\infty)}))$$

and it follows then from (3.56), (3.57), and Corollary 3.2.6 that

$$\begin{aligned} \alpha_1 \eta_1^{(\infty)} &= \pi_{V_2}(-\operatorname{div}(\alpha_1 \xi_1)) = \pi_{V_2} P_{\beta_1 K}(-u^{(\infty)} - \operatorname{div}(\alpha_1 \xi_1)) && (3.59) \\ \alpha_2 \eta_2^{(\infty)} &= \pi_{V_1}(-\operatorname{div}(\alpha_2 \xi_2)) = \pi_{V_1} P_{\beta_2 K}(-u^{(\infty)} - \operatorname{div}(\alpha_2 \xi_2)). && (3.60) \end{aligned}$$

Plugging (3.59) and (3.60) in (3.58) and using the definition of $z_1^{(\infty)}$ and $z_2^{(\infty)}$, see (3.49) and (3.50), yields

$$\begin{aligned} 0 &= -T^*(Tu^{(\infty)} - g) - \operatorname{div}(\alpha_1 \xi_1) - \operatorname{div}(\alpha_2 \xi_2) + (\pi_{V_2} \operatorname{div}(\alpha_1 \xi_1) + \pi_{V_1} \operatorname{div}(\alpha_2 \xi_2)) \\ &= -T^*(Tu^{(\infty)} - g) - (\pi_{V_1} \operatorname{div}(\alpha_1 \xi_1) + \pi_{V_2} \operatorname{div}(\alpha_2 \xi_2)). \end{aligned}$$

Therefore, if there exists ξ such that $\operatorname{div}(\xi) \in \alpha \partial |\nabla u^{(\infty)}|(\Omega)$ and

$$\operatorname{div}(\xi) = \pi_{V_1} \operatorname{div}(\alpha_1 \xi_1) + \pi_{V_2} \operatorname{div}(\alpha_2 \xi_2), \tag{3.61}$$

then ξ also satisfies

1. $\|\xi\|_\infty \le \alpha$,
2. $\langle \xi(x), \nabla u^{(\infty)}(x)\rangle_{\mathbb{R}^d} + \alpha|\nabla u^{(\infty)}|(x) = 0$ for all $x \in \Omega$,
3. $T^*\xi_0 - \operatorname{div}(2\xi) = 0$,
4. $-\xi_0 = 2(Tu^{(\infty)} - g)$.

The existence of such ξ is a necessary and sufficient condition for $u^{(\infty)}$ to be a minimizer by Proposition 3.2.2. Then ξ satisfies

$$-u^{(\infty)} + z^{(\infty)} = T^*(g - Tu^{(\infty)}) = -\alpha_1 u_1^{(\infty)} + \alpha_1 z_1^{(\infty)} - \alpha_2 u_2^{(\infty)} + \alpha_2 z_2^{(\infty)} = \operatorname{div}(\xi),$$

that is,

$$-z^{(\infty)} = -u^{(\infty)} - \operatorname{div}(\xi) \quad \text{and} \quad \operatorname{div}(\xi) \in \alpha|\nabla u^{(\infty)}|(\Omega),$$

where $z^{(\infty)} := u^{(\infty)} + T^*(g - Tu^{(\infty)})$. Note, that for $i = 1, 2$,

$$z_i^{(\infty)} \neq \pi_{V_i} z^{(\infty)}, \quad \text{and} \quad -\alpha_i u_i^{(\infty)} + \alpha_i z_i^{(\infty)} = -u_i^{(\infty)} + \pi_{V_i} z^{(\infty)}.$$

By $\operatorname{div}(\xi) \in \alpha\partial|\nabla u^{(\infty)}|(\Omega)$ and Corollary 3.2.6, this is equivalent to

$$u^{(\infty)} - z^{(\infty)} = -\operatorname{div}(\xi) = P_{\alpha K}(-u^{(\infty)} - \operatorname{div}(\xi)) = P_{\alpha K}(-z^{(\infty)}).$$

Hence,

$$-u^{(\infty)} = (I - P_{\alpha K})(-z^{(\infty)}) = \arg\min_u \|u + z^{(\infty)}\|_2^2 + 2\alpha|\nabla u|(\Omega),$$

which proves the theorem.

The proof of Theorem 3.2.4 provides us with another characterization of $u^{(\infty)}$ being a minimizer of (3.24) by ξ_1, ξ_2 in (3.56), (3.57).

Corollary 3.2.7. *Let $\alpha_1 \le 1$, $\alpha_2 \le 1$. The limit $u^{(\infty)}$, obtained in Theorem 3.2.4 b), is a minimizer of* (3.24) *if and only if there exist ξ_1, ξ_2 in* (3.56), (3.57) *with* $\operatorname{div}(\alpha_1\xi_1) = \operatorname{div}(\alpha_2\xi_2)$.

Proof. First let us prove the statement for $\alpha_1 = \alpha_2 = 1$: if $u^{(\infty)}$ is a minimizer of (3.24), then Theorem 3.2.4 and Remark 3.1.2 say that

$$u^{(\infty)} = (I - P_{\alpha K})(z^{(\infty)}).$$

Since $\alpha_1 = \alpha_2 = 1$, we obtain

$$z_1^{(\infty)} = \pi_{V_1} z^{(\infty)}, \quad z_2^{(\infty)} = \pi_{V_2} z^{(\infty)}.$$

We then can rephrase this in two different ways as follows:

$$\begin{aligned} u_1^{(\infty)} &= (I - P_{\alpha K})(z_1^{(\infty)} + u_2^{(\infty)} - (u_2^{(\infty)} - z_2^{(\infty)})) - u_2^{(\infty)}, \\ \text{or} \quad u_2^{(\infty)} &= (I - P_{\alpha K})(z_2^{(\infty)} + u_1^{(\infty)} - (u_1^{(\infty)} - z_1^{(\infty)})) - u_1^{(\infty)}. \end{aligned}$$

By Theorem 3.1.4, we can take

$$\eta_1^{(\infty)} = u_2^{(\infty)} - z_2^{(\infty)}, \quad \eta_2^{(\infty)} = u_1^{(\infty)} - z_1^{(\infty)}.$$

This implies $\mathrm{div}(\xi_1) = \mathrm{div}(\xi_2)$ from (3.56) and (3.57). On the other hand, if $\mathrm{div}(\xi_1) = \mathrm{div}(\xi_2)$, then (3.61) implies that $u^{(\infty)}$ is a minimizer of (3.24).

Now let us prove the statement for $\alpha_1, \alpha_2 \leq 1$: suppose that $u^{(\infty)}$ is a minimizer of (3.24). Then Theorem 3.2.4 says that

$$\begin{gathered} u^{(\infty)} = (I - P_{\alpha K})(z^{(\infty)}) \\ \text{if and only if} \\ \mathrm{div}(\xi) = -u^{(\infty)} + z^{(\infty)} \in \alpha\partial|\nabla u^{(\infty)}|(\Omega) \text{ for some } \xi. \end{gathered}$$

By the above considerations, we know that there exist $\eta_1^{\infty,1}, \eta_2^{\infty,1}$ such that

$$\eta_1^{\infty,1} = u_2^{(\infty)} - \pi_{V_2} z^{(\infty)} = \alpha_2 u_2^{(\infty)} - \alpha_2 z_2^{(\infty)}, \quad \eta_2^{\infty,1} = u_1^{(\infty)} - \pi_{V_1} z^{(\infty)} = \alpha_1 u_1^{(\infty)} - \alpha_1 z_1^{(\infty)}$$

and

$$-u_1^{(\infty)} + (\pi_{V_1} z^{(\infty)} - \eta_1^{\infty,1}) = \mathrm{div}(\xi) = -u_2^{(\infty)} + (\pi_{V_2} z^{(\infty)} - \eta_2^{\infty,1}).$$

Let $\eta_i^{\infty,\alpha_i} = \frac{\eta_i^{\infty,1}}{\alpha_i}$, $\xi_i^{\alpha_i} = \frac{\xi}{\alpha_i}$ for $i = 1, 2$. Then

$$\begin{aligned} \mathrm{div}(\xi_1^{\alpha_1}) &= -u_1^{(\infty)} + (z_1^{(\infty)} - \eta_1^{\infty,\alpha_1}), \\ \mathrm{div}(\xi_2^{\alpha_2}) &= -u_2^{(\infty)} + (z_2^{(\infty)} - \eta_2^{\infty,\alpha_2}). \end{aligned}$$

Moreover one can see that $\mathrm{div}(\xi_i^{\alpha_i}) \in \beta_i \partial|\nabla u^{(\infty)}|(\Omega)$ for $i = 1, 2$. Hence if we let $\xi_1 = \xi_1^{\alpha_1}$ and $\xi_2 = \xi_2^{\alpha_2}$, then $\mathrm{div}(\alpha_1 \xi_1) = \mathrm{div}(\xi) = \mathrm{div}(\alpha_2 \xi_2)$.

On the other hand, if there exist ξ_1, ξ_2 satisfying $\mathrm{div}(\alpha_1\xi_1) = \mathrm{div}(\alpha_2\xi_2)$ in (3.56), (3.57), then by (3.61), we know that the limit $u^{(\infty)}$ is a minimizer of (3.24). □

3.2.6 Numerical Validation

In this section we illustrate the performance of the algorithm in (3.13) for the minimization of (3.24) when T is a blur operator with *averaging* kernel κ supported on 3×3 pixels and uniform values $1/9$. The function space is split into $\mathcal{N} \in \mathbb{N}$ orthogonal spaces by a wavelet space decomposition such that

$$\mathcal{H} = \mathcal{V}_{2-\mathcal{N}} \oplus \left(\bigoplus_{j=2-\mathcal{N}}^{0} \mathcal{W}_j \right).$$

In addition, we set $V_1 := \mathcal{V}_{2-\mathcal{N}}$ and $V_i := \mathcal{W}_{2-i}$ for $i = 2, 3, \ldots, \mathcal{N}$. Note that for $\mathcal{N} = 1$ we have that $V_1 = \mathcal{H}$ and thus we have no splitting. In order to gain maximal performance, the preconditioner constants are always chosen as

$$\alpha_i = \|T_{V_i} S_i\|^2, \tag{3.62}$$

for $i = 1, \dots, \mathcal{N}$, as already discussed in detail for $\mathcal{N} = 2$ in Section 3.2.3.

In our numerical examples we only consider decompositions by using Haar wavelets. In this case it is easy to see that the preconditioner constant for the scale space $\mathcal{V}_{2-\mathcal{N}}$ is simply $\alpha_1 = \|T\|$ and the preconditioner constants for the wavelet spaces $\mathcal{W}_j$, $j = 0, \dots, 2 - \mathcal{N}$, are strictly smaller than the norm of T.

The implementation of the algorithm is done as suggested and discussed in [66]. That is the subiterations in (3.13) are solved by computing the minimizers by means of oblique thresholding, cf. Theorem 3.1.4.

Computation of the Oblique Thresholding (OT)

To solve the subiterations in (3.13) we compute the minimizer by means of oblique thresholding. More precisely, let us denote $u_2 = u_2^{(n)}$, $u_1 = u_1^{(n+1,\ell+1)}$, and $z_1 = u_1^{(n+1,\ell)} + \frac{1}{\alpha_1}\pi_{V_1} T^*(g - Tu_2 - Tu_1^{(n+1,\ell)})$. We shall compute the minimizer u_1 of the first subminimization problem by

$$u_1 = (I - P_{\beta_1 K})(z_1 + u_2 - \eta_1) - u_2 \in V_1$$

for an $\eta_1 \in V_2$, which fulfills

$$\eta_1 = \pi_{V_2} P_{\beta_1 K}(\eta_1 - z_1 - u_2).$$

Hence the element $\eta_1 \in V_2$ is a limit of the corresponding fixed point iteration

$$\eta_1^{(0)} \in V_2, \quad \eta_1^{(m+1)} = \pi_{V_2} P_{\beta_1 K}(\eta_1^{(m)} - z_1 - u_2), \ m \geq 0. \tag{3.63}$$

Here K is defined as in Section 3.2.2, i.e.,

$$K = \left\{ \operatorname{div} p : p \in \mathcal{H}^d, |p(x)|_\infty \leq 1 \quad \text{for all } x \in \Omega \right\}.$$

To compute the orthogonal projection onto $\beta_1 K$ in the oblique thresholding we use an algorithm proposed by Chambolle in [23]. His algorithm is based on considerations of the convex conjugate of the total variation and on exploiting the corresponding optimality condition. It amounts to compute $P_{\beta_1 K}(g)$ approximately by $\beta_1 \operatorname{div} p^{(n)}$, where $p^{(n)}$ is the n-th iterate of the following semi-implicit gradient descent algorithm:

Choose $\tau > 0$, let $p^{(0)} = 0$ and, for any $n \geq 0$, iterate

$$p^{(n+1)}(x) = \frac{p^{(n)}(x) + \tau(\nabla(\operatorname{div} p^{(n)} - g/\beta_1))(x)}{1 + \tau \left|(\nabla(\operatorname{div} p^{(n)} - g/\beta_1))(x)\right|}.$$

For $\tau > 0$ sufficiently small, i.e., $\tau < 1/8$, the iteration $\beta_1 \operatorname{div} p^{(n)}$ was shown to converge to $P_{\beta_1 K}(g)$ as $n \to \infty$ (compare [23, Theorem 3.1]). Let us stress that we propose here this algorithm just for the ease of its presentation and its implementation; its choice for the approximation of projections is of course by no means a restriction and one may want to implement other recent, and perhaps faster strategies, e.g., [26, 46, 69, 96, 119].

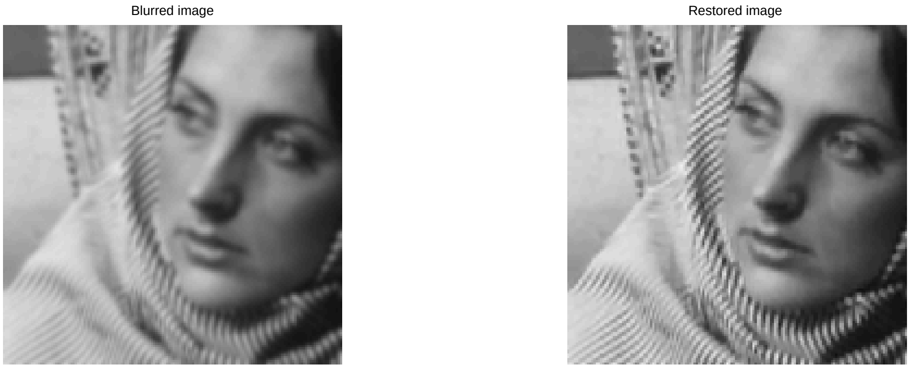
Experiments

The experiments we include in this chapter are produced in Matlab with the implementation described above of the algorithm in (3.13). In our examples we stop the algorithm as soon as the energy $\mathcal{J}$ reaches a significant level, i.e.,

$$\mathcal{J}(u^*) \leq \epsilon, \tag{3.64}$$

where u^* denotes the first iterate for which (3.64) is fulfilled and ϵ is an estimate of the minimal energy.

In Figure 3.2 we show an image of size 156×156 pixels, which was corrupted by the blur operator T as above. In order to deblur this image we split the function space of the image into orthogonal subspaces via a wavelet space decomposition and compute its solution by the algorithm with $\alpha = 10^{-5}$ and stopping criterion (3.64) with $\epsilon = 0.04$. The computed result for 4 subspaces is shown in Figure 3.2 on the right hand side.



Blurred image

Restored image

Figure 3.2: *On the left we depict an image, blurred with an averaging kernel. On the right we show the corresponding solution computed on 4 orthogonal subspaces by the algorithm in* (3.13) *with* $\alpha = 10^{-5}$ *and stopping criterion* (3.64) *with* $\epsilon = 0.04$.

$\mathcal{N}$	1	2	3	4	5	6
Iterations	525	12	8	6	6	6
CPU (s)	40.80	2.59	2.37	2.34	3.29	4.05

Table 3.1: *Performance of the wavelet decomposition algorithm in* (3.13) *for image deblurring (uniform kernel) with energy-stopping criterion* (3.64) *with* $\epsilon = 0.04$*: the number of iterations and CPU time in seconds are shown with respect to the number* $\mathcal{N}$ *of subspace decompositions.*

In the same setting as above, we solve this specific deblurring problem with the algorithm in (3.13) for different numbers of subspaces and compare its performance with respect to the needed iterations and computational time in Table 3.1. Note that for $\mathcal{N} = 1$ we solve this problem without any decomposition on the whole space $\mathcal{H}$. We see in Table 3.1 that the performance in this case is clearly the worst. When we solve the same problem with a decomposition into two or more wavelet spaces only a very few iterations are needed to reach the stopping criterion. Additionally, a decomposition into only 2 subspaces leads to a significant speed-up concerning the computational time, cf. also Figure 3.5.

By using Lemma 3.2.7 we check for a splitting into 2 orthogonal subspaces whether the sequential algorithm numerically converges to a minimizer by looking at

$$\| \operatorname{div}(\alpha_1 \xi_1^{(n)}) - \operatorname{div}(\alpha_2 \xi_2^{(n)}) \|, \tag{3.65}$$

where

$$\begin{aligned} \operatorname{div}(\xi_1^{(n)}) &= -u_1^{(n)} + (z_1^{(n)} - \eta_1^{(n)}) \\ \operatorname{div}(\xi_2^{(n)}) &= -u_2^{(n)} + (z_2^{(n)} - \eta_2^{(n)}). \end{aligned}$$

In Figure 3.3 we plot the decay of this norm discrepancy, indicator of the distance from convergence to a minimizer, with respect to the iterations n. The indicator numerically converges to zero for n increasing and the algorithm numerically converges to a minimizer of the original problem.

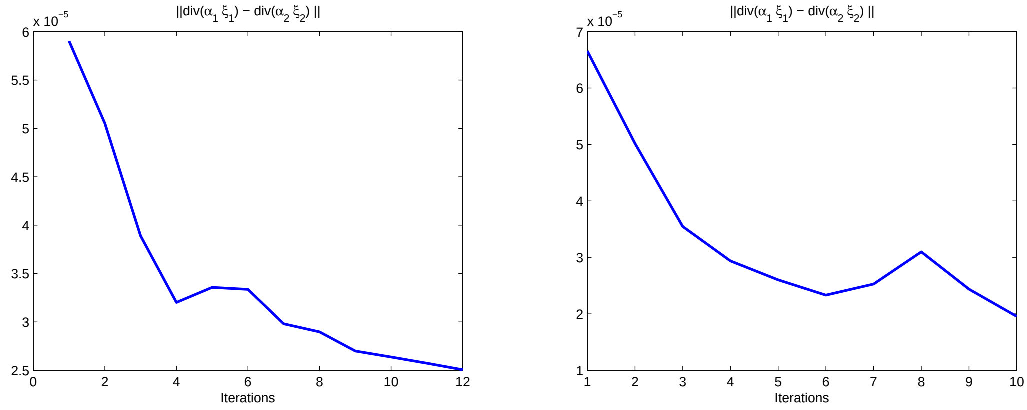


Figure 3.3: *We plot* $\| \operatorname{div}(\alpha_1 \xi_1^{(n)}) - \operatorname{div}(\alpha_2 \xi_2^{(n)}) \|$ *for the problem of Figure 3.2 (left) and Figure 3.4 (right) in view of Lemma 3.2.7 in order to check whether the algorithm is indeed converging.*

In Figure 3.4 we depict another example of an image deblurring problem, where the image of size 279 × 285 pixels was blurred by the averaging kernel from above. The image is again recovered via the algorithm in (3.13) by splitting the function space $\mathcal{H}$ into orthogonal wavelet spaces. We take as the stopping criterion (3.64) with $\epsilon = 0.058$ and as a regularization parameter $\alpha = 10^{-5}$. In Table 3.2 we show the behaviour of the algorithm for different numbers of subspaces.

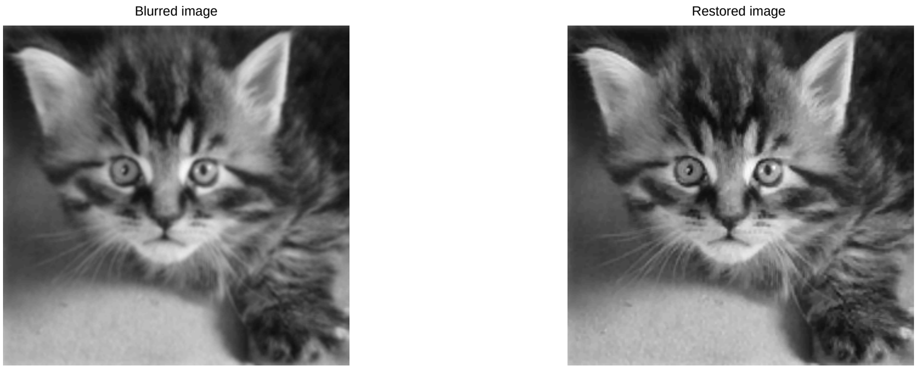


Figure 3.4: *On the left we show an image, blurred by an averaging kernel. On the right we show the corresponding solution computed alternating on 3 orthogonal subspaces by the algorithm in* (3.13) *with* $\alpha = 10^{-5}$ *and stopping criterion* (3.64) *with* $\epsilon = 0.058$.

$\mathcal{N}$	1	2	3	4	5	6
Iterations	405	10	8	7	7	7
CPU (s)	86.94	8.37	11.37	13.75	17.59	22.24

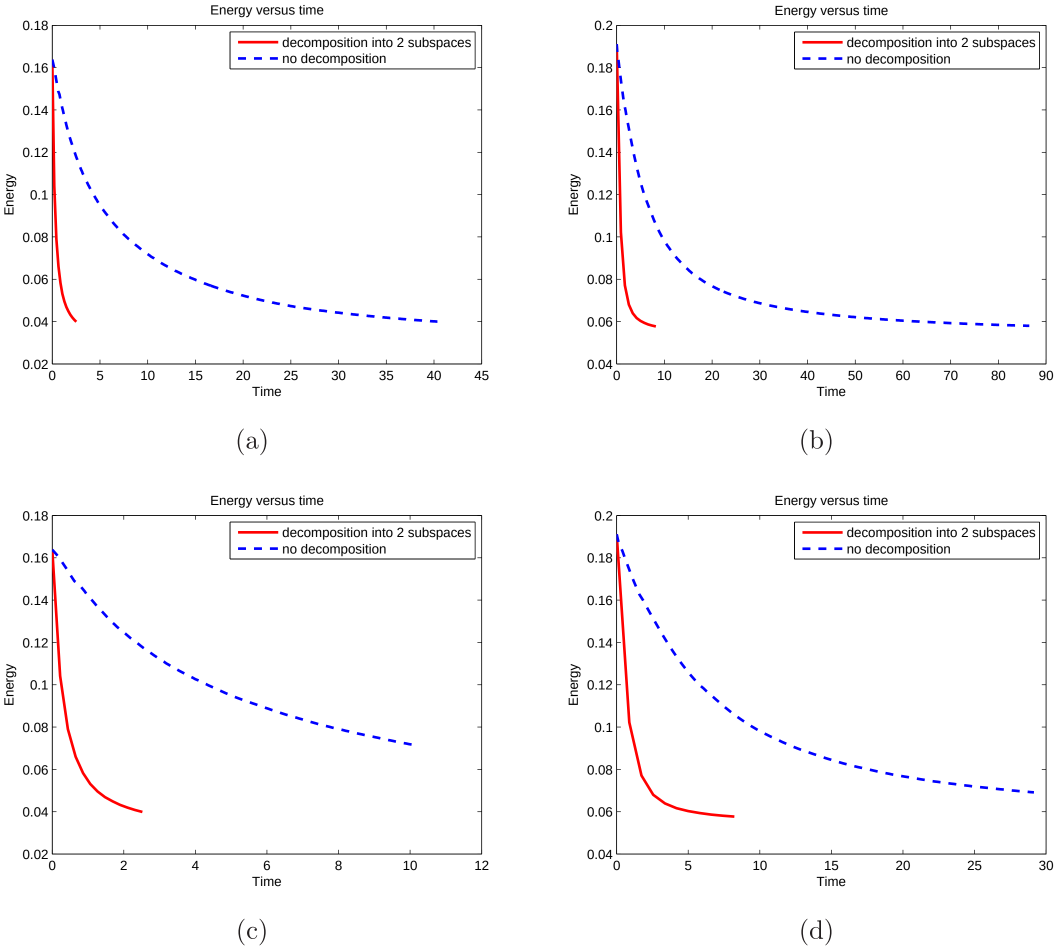
Table 3.2: *Performance of the wavelet decomposition algorithm in* (3.13) *for image deblurring (uniform kernel) with energy-stopping criterion* (3.64) *with* $\epsilon = 0.058$*: the number of iterations and CPU time in seconds are shown with respect to the number* $\mathcal{N}$ *of subspace decompositions.*

Again we see from the numerical results that with a decomposition into 2 subspaces the speed of convergence increases dramatically as depicted in Figure 3.5.

Let us display in Figure 3.6 also the “distance” between the obtained estimate and the original image. Therefore we recall the definition of *Signal-to-Error-Ratio Gain* [116] given by

$$\mathrm{SERG} = 20 \log_{10} \frac{\|g - u_{org}\|}{\|u^* - u_{org}\|},$$

where u_{org} denotes the original image before blurring. In Figure 3.6 we show the evolution of this measure with respect to the time need for both mentioned deblurring problems for $\mathcal{N} = 1$ (no splitting) and for $\mathcal{N} = 2$ (splitting into 2 subspaces).



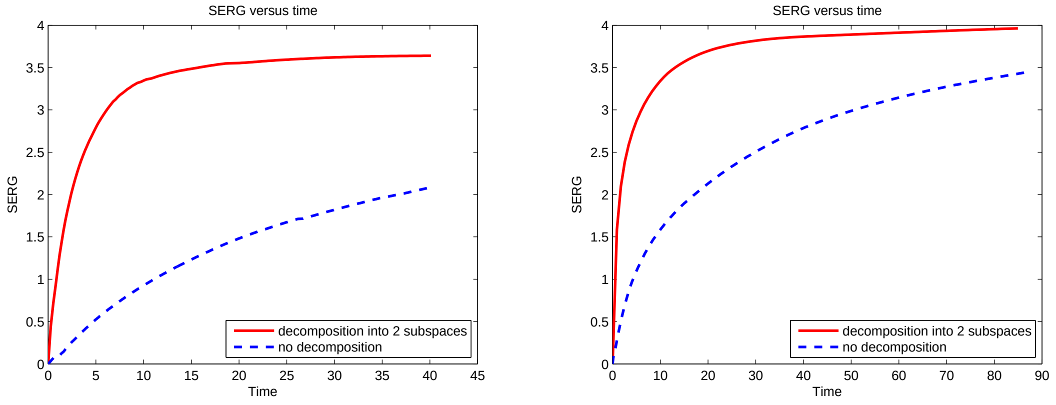
(a)

(b)

(c)

(d)

Figure 3.5: *We show for $\mathcal{N} = 1$ and $\mathcal{N} = 2$ the decay of the energy. For the deblurring problem of Figure 3.2 the decays are plotted in (a) and zoomed in to the first 10 seconds in (c). We depict the decays of the energies for the deblurring problem of Figure 3.4 in (b) and zoomed in to the first 25 seconds in (d).*



(a)

(b)

Figure 3.6: *We show for* $\mathcal{N} = 1$ *and* $\mathcal{N} = 2$ *the evolution of the quality measure SERG: in (a) for the deblurring problem of Figure 3.2 and in (b) for the deblurring problem of Figure 3.4.*

Chapter 4

Domain Decomposition for Total Variation Minimization

This chapter is dedicated to overlapping and non-overlapping domain decomposition methods for total variation minimization. In order to successfully show convergence of these methods we have to take technical issues into account, which we describe in Section 4.1. In Section 4.2 we will introduce and investigate sequential and parallel overlapping domain decomposition methods [64]. In Section 4.3 we specify the subspace correction method recalled in Section 3.1 to a non-overlapping domain decomposition method for total variation minimization, as already proposed in [66], and we show that in a discrete setting the algorithm always converges to an expected minimizer, as reported in [75] and inspired by the convergence proof of the overlapping domain decomposition method presented in Section 4.2. The subminimization problems in the overlapping and non-overlapping domain decomposition methods are solved by the iterative oblique thresholding, see Theorem 3.1.4, which is based on an iterative proximity map algorithm and the computation of a Lagrange multiplier by a fixed point iteration. In Section 4.4 we report on a more efficient approach as appearing in [76].

4.1 Technical Issues

Our analysis of the proposed domain decomposition methods is performed, as in Section 3.2, for a discrete approximation of the continuous functional (1.4), which we denote for ease again by $\mathcal{J}$ in (4.1). In Section 3.2.1 we summerized the main properties and advantages of such a discrete approximation. In particular, we noted, that the discrete approach has the virtues of being practical for numerical implementations, of correctly approximating the continuous setting, and of retaining the major features, which makes the problem interesting. Let us point out that in this section we use traces of bounded variation of functions, which are not continuous with respect to the weak-$*$-topology of BV. However, they are indeed continuous in a discrete setting, which has only strong topology. Therefore the discrete setting eventually provides us with a framework in which we are able to prove successfully in Theorem 4.2.8, Theorem 4.2.12, Theorem 4.3.1, and Theorem 4.3.4 that sequential and parallel domain decomposition methods for total variation minimization indeed converge to minimizers of $\mathcal{J}$, which constitute the main result of this chapter. In light of the counterexamples mentioned in the previous chapter, we reiterate the relevance of this result. Nevertheless, it is important to mention that there are also other very recent attempts of addressing domain decomposition strategies to functionals with a total variation constraint, which claim to work sufficiently well in practice, but without any sufficient theoretical convergence analysis [88, 121]. One particular interesting approach was developed in the diploma thesis of Jahn Müller [88]. There a primal-dual Newton method was used in order to solve the subminimization problem on each subdomain. The numerical results look very promising and show a significant computational speed up. However, it was not possible to show that the algorithm is indeed converging to the expected minimizer. Another domain decomposition strategy for total variation appeared in [121], but in order to obtain a convergence result it was needed to regularize the problem.

For ease of presentation, and in order to avoid unnecessary technicalities, we limit our analysis to splitting the problem into two subdomains Ω_1 and Ω_2. This is by no means a restriction. The generalization to multiple domains comes quite natural in our specific setting, see also Remark 4.2.10.

Since we are interested in a discrete setting we use throughout this chapter the notations provided in Section 3.2.2.

4.2 The Overlapping Domain Decomposition Algorithm

Similar as in Section 3.2, we are interested in the minimization of the functional

$$\mathcal{J}(u) := \|Tu - g\|_2^2 + 2\alpha \left|\nabla(u)\right|(\Omega), \tag{4.1}$$

where $T : \mathcal{H} \to \mathcal{H}$ is now any bounded linear operator, $g \in \mathcal{H}$ is a datum, and $\alpha > 0$ is a fixed constant. We recall that in order to guarantee the existence of minimizers for (4.1) we assume condition (**C**), i.e., that $\mathcal{J}$ is coercive in $\mathcal{H}$.

Now, instead of minimizing (4.1) on the whole domain, we decompose Ω into two overlapping subdomains Ω_1 and Ω_2 such that $\Omega = \Omega_1 \cup \Omega_2$, $\Omega_1 \cap \Omega_2 \neq \emptyset$, and a certain splitting property for the total variation, i.e.,

$$\begin{aligned} |\nabla u|(\Omega) &= |\nabla u_{|\Omega_1}|(\Omega_1) + c_1(u_{|(\Omega_2 \setminus \Omega_1)\cup\Gamma_1}), \\ |\nabla u|(\Omega) &= |\nabla u_{|\Omega_2}|(\Omega_2) + c_2(u_{|(\Omega_1 \setminus \Omega_2)\cup\Gamma_2}), \end{aligned} \tag{4.2}$$

where c_1 and c_2 are suitable functions that depend only on the restrictions $u_{|(\Omega_2 \setminus \Omega_1)\cup\Gamma_1}$ and $u_{|(\Omega_1 \setminus \Omega_2)\cup\Gamma_2}$ respectively, is fulfilled. Note that formula (4.2) is the discrete analogous of (2.6) in the continuous setting. The simplest examples of discrete domains with such a property are discrete d-dimensional rectangles (*d-orthotopes*). For instance, with our notations from Section 3.2.2, it is easy to check that for $d = 1$ and for Ω being a discrete interval, one computes $c_1(u_{|(\Omega_2\setminus\Omega_1)\cup\Gamma_1}) = |\nabla u_{|(\Omega_2\setminus\Omega_1)\cup\Gamma_1}|((\Omega_2 \setminus \Omega_1) \cup \Gamma_1)$, $c_2(u_{|(\Omega_1\setminus\Omega_2)\cup\Gamma_2}) = |\nabla u_{|(\Omega_1\setminus\Omega_2)\cup\Gamma_2}|((\Omega_1 \setminus \Omega_2) \cup \Gamma_2))$; it is straightforward to generalize the computation to $d > 1$. Hence, for ease of presentation, we will assume to work with d-orthotope domains, also noting that such decompositions are already sufficient for any practical use in image processing, and stressing that the results can be generalized also to subdomains with different shapes as long as (4.2) is satisfied. However, for consistency of the definitions of gradient and divergence, we assume that also the subdomains Ω_i are discrete d-orthotopes as well as Ω, stressing that this is by no means a restriction, but only for ease of presentation. Due to this overlapping decomposition of the domain Ω, the function space $\mathcal{H}$ is split into two closed subspaces $V_j = \{u \in \mathcal{H} : \operatorname{supp}(u) \subset \Omega_j\}$, for $j = 1, 2$. Note that $\mathcal{H} = V_1 + V_2$ is not a direct sum of V_1 and V_2, but just a linear sum of subspaces. Thus any $u \in \mathcal{H}$ has a nonunique representation

$$u(x) = \begin{cases} u_1(x) & x \in \Omega_1 \setminus \Omega_2 \\ u_1(x) + u_2(x) & x \in \Omega_1 \cap \Omega_2 \\ u_2(x) & x \in \Omega_2 \setminus \Omega_1 \end{cases}, \quad u_i \in V_i, \quad i = 1, 2. \tag{4.3}$$

We denote by Γ_1 the interface between Ω_1 and $\Omega_2 \setminus \Omega_1$ and by Γ_2 the interface between Ω_2 and $\Omega_1 \setminus \Omega_2$ (the interfaces are naturally defined in the discrete setting).

We introduce the trace operator of the restriction to a boundary Γ_i

$$\operatorname{Tr}_{|\Gamma_i} : V_i \to \mathbb{R}^{\Gamma_i}, \quad i = 1, 2$$

with $\operatorname{Tr}_{|\Gamma_i} v_i = v_{i|\Gamma_i}$, the restriction of v_i on Γ_i. Note that $\mathbb{R}^{\Gamma_i}$ is as usual the set of maps from Γ_i to $\mathbb{R}$. The trace operator is clearly a linear and continuous operator. We additionally fix a *bounded uniform partition of unity* (BUPU) $\{\chi_1, \chi_2\} \subset \mathcal{H}$ such that

(a) $\operatorname{Tr}_{|\Gamma_i} \chi_i = 0$ for $i = 1, 2$,

(b) $\chi_1 + \chi_2 = 1$,

(c) $\operatorname{supp} \chi_i \subset \Omega_i$ for $i = 1, 2$,

(d) $\max\{\|\chi_1\|_\infty, \|\chi_2\|_\infty\} = c_\chi < \infty$.

We would like to solve

$$\arg\min_{u\in\mathcal{H}} \mathcal{J}(u) \tag{4.4}$$

by picking an initial $V_1 + V_2 \ni \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} := u^{(0)} \in \mathcal{H}$, e.g., $\tilde{u}_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} u_1^{(n+1)} \approx \arg\min_{\substack{v_1 \in V_1 \\ \mathrm{Tr}_{|\Gamma_1} v_1 = 0}} \mathcal{J}(v_1 + \tilde{u}_2^{(n)}) \\ u_2^{(n+1)} \approx \arg\min_{\substack{v_2 \in V_2 \\ \mathrm{Tr}_{|\Gamma_2} v_2 = 0}} \mathcal{J}(u_1^{(n+1)} + v_2) \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)} \\ \tilde{u}_1^{(n+1)} := \chi_1 \cdot u^{(n+1)} \\ \tilde{u}_2^{(n+1)} := \chi_2 \cdot u^{(n+1)}. \end{cases} \tag{4.5}$$

Note that we are minimizing over functions $v_i \in V_i$ for $i = 1, 2$ that vanish on the interior boundaries, i.e., $\mathrm{Tr}_{|\Gamma_i} v_i = 0$. Moreover $u^{(n)}$ is the sum of the local minimizers $u_1^{(n)}$ and $u_2^{(n)}$, which are not uniquely determined on the overlapping part. Therefore we introduced a suitable correction by χ_1 and χ_2 in order to force the subminimizing sequences $(u_1^{(n)})_n$ and $(u_2^{(n)})_n$ to remain uniformly bounded. This issue will be explained in detail below, see Lemma 4.2.6. From the definition of χ_i, $i = 1, 2$, it is clear that

$$u_1^{(n+1)} + u_2^{(n+1)} = u^{(n+1)} = (\chi_1 + \chi_2) u^{(n+1)} = \tilde{u}_1^{(n+1)} + \tilde{u}_2^{(n+1)}.$$

Note that in general $u_1^{(n)} \neq \tilde{u}_1^{(n)}$ and $u_2^{(n)} \neq \tilde{u}_2^{(n)}$. The realization of the approximate solution to the individual subspace minimizations, discussed in the next section, follows basically the same idea as the one provided in Section 3.1.1 for the general subspace correction algorithm in (3.2).

4.2.1 Local Minimization by Lagrange Multipliers

Let us consider, for example, the subspace minimization on Ω_1

$$\arg \min_{\substack{v_1 \in V_1 \\ \mathrm{Tr}_{|\Gamma_1} v_1 = 0}} \mathcal{J}(v_1 + u_2) = \arg \min_{\substack{v_1 \in V_1 \\ \mathrm{Tr}_{|\Gamma_1} v_1 = 0}} \|T v_1 - (g - T u_2)\|_2^2 + 2\alpha |\nabla(v_1 + u_2)| (\Omega). \tag{4.6}$$

We observe that $\left\{ u \in \mathcal{H} : \mathrm{Tr}_{|\Gamma_1} u = \mathrm{Tr}_{|\Gamma_1} u_2, \ \mathcal{J}(u) \leq C \right\} \subset \{\mathcal{J} \leq C\}$. By assumption (**C**) these sets are bounded and hence the minimization problem (4.6) has solutions.

In order to realize an approximate solution to (4.6) we use the following algorithm: for $u_1^{(0)} = \tilde{u}_1^{(0)} \in V_1$,

$$u_1^{(\ell+1)} = \arg \min_{\substack{u_1 \in V_1 \\ \mathrm{Tr}_{|\Gamma_1} u_1 = 0}} \mathcal{J}_1^s(u_1 + u_2, u_1^{(\ell)}), \quad \ell \geq 0, \tag{4.7}$$

where $\mathcal{J}_1^s$ is the surrogate functional of $\mathcal{J}$ defined as in (3.4), i.e., for $a, u_1 \in V_1$, $u_2 \in V_2$ we have

$$\mathcal{J}_1^s(u_1 + u_2, a) := \mathcal{J}(u_1 + u_2) + \|u_1 - a\|_2^2 - \|T(u_1 - a)\|_2^2. \tag{4.8}$$

Note that $\mathcal{J}_1^s$ can be written in the following form

$$\mathcal{J}_1^s(u_1+u_2,a) = \|u_1-(a+(T^*(g-Tu_2-Ta))_{|\Omega_1})\|_2^2+2\alpha\,|\nabla(u_1+u_2)|\,(\Omega)+\Phi(a,g,u_2),$$

with Φ being a function of a, g, u_2 only. Additionally in (4.7) we can restrict the total variation on Ω_1 only, since we have

$$|\nabla(u_1+u_2)|\,(\Omega) \;=\; \left|\nabla(u_1+u_2)_{|\Omega_1}\right|(\Omega_1)+c_1(u_{2|_{(\Omega_2\setminus\Omega_1)\cup\Gamma_1}}), \tag{4.9}$$

where we used (4.2) and the assumption that u_1 vanishes on the interior boundary Γ_1. Hence (4.7) is equivalent to

$$\arg\min_{\substack{u_1\in V_1\\ \mathrm{Tr}_{|\Gamma_1}u_1=0}} \mathcal{J}_1^s(u_1+u_2,u_1^{(\ell)}) = \arg\min_{\substack{u_1\in V_1\\ \mathrm{Tr}_{|\Gamma_1}u_1=0}} \|u_1-z_1\|_2^2+2\alpha\left|\nabla(u_1+u_2)_{|\Omega_1}\right|(\Omega_1), \tag{4.10}$$

where $z_1 = u_1^{(\ell)} + (T^*(g - Tu_2 - Tu_1^{(\ell)}))_{|\Omega_1}$. Similarly the same arguments work for the second subproblem.

Let us now clarify how to practically compute $u_1^{(\ell+1)}$ for a given $u_1^{(\ell)}$. To do so we need to recall a useful result from convex analysis.

We observe that in order to solve the subminimization problems (4.10) we have to solve a constrained minimization problem of the type (3.6), i.e.,

$$\arg\min_{x\in\mathcal{H}}\{F(x) : Gx = 0\}, \tag{4.11}$$

where $F : \mathcal{H} \to \mathbb{R}$ is a convex functional and $G : \mathcal{H} \to \mathcal{H}$ is a bounded linear operator on $\mathcal{H}$. We have the following useful result, which is the discrete version of Theorem 3.1.3.

Theorem 4.2.1. ([71, Theorem 2.1.4, p. 305]). *Let* $N = \{G^*\lambda : \lambda\in\mathcal{H}\} = \mathrm{Range}(G^*)$. *Then,* $x_0 \in \{x\in\mathcal{H} : Gx = 0\}$ *solves the constrained minimization problem* (4.11) *if and only if*

$$0 \in \partial F(x_0) + N.$$

Oblique Thresholding (OT)

We want to exploit Theorem 4.2.1 in order to produce an algorithmic solution to each iteration step (4.7), which practically stems from the solution of a problem of this type

$$\arg\min_{\substack{u_1\in V_1\\ \mathrm{Tr}_{|\Gamma_1}u_1=0}} \|u_1-z_1\|_2^2+2\alpha\left|\nabla(u_1+u_2)_{|\Omega_1}\right|(\Omega_1).$$

It is well-known how to solve this problem if $u_2 \equiv 0$ in Ω_1 and if the trace condition is not imposed. For the general case we propose to use the oblique thresholding strategy, which was already introduced for the general subspace correction method in Section 3.1. We recall in the following theorem the main idea of this technique for our specific setting (cf. Theorem 3.1.4). In what follows all the involved quantities are restricted to Ω_1, e.g., $u_2 = u_{2|\Omega_1}$.

Theorem 4.2.2 (Oblique thresholding). *For $u_2 \in V_2$ and for $z_1 \in V_1$ the following statements are equivalent:*

(i) $u_1^* = \arg\min_{\substack{u_1 \in V_1 \\ \mathrm{Tr}_{|\Gamma_1} u_1 = 0}} \|u_1 - z_1\|_2^2 + 2\alpha \left|\nabla(u_1 + u_2)\right|(\Omega_1)$*;*

(ii) there exists $\eta \in \mathrm{Range}(\mathrm{Tr}_{|\Gamma_1})^* = \{\eta \in V_1$ *with* $\mathrm{supp}(\eta) = \Gamma_1\}$ *such that* $0 \in u_1^* - (z_1 - \eta) + \alpha \partial_{V_1} \left|\nabla(\cdot + u_2)\right|(\Omega_1)(u_1^*)$*;*

(iii) there exists $\eta \in V_1$ *with* $\mathrm{supp}(\eta) = \Gamma_1$ *such that* $u_1^* = (I - P_{\alpha K})(z_1 + u_2 - \eta) - u_2 \in V_1$ *and* $\mathrm{Tr}_{|\Gamma_1} u_1^* = 0$*;*

(iv) there exists $\eta \in V_1$ *with* $\mathrm{supp}(\eta) = \Gamma_1$ *such that* $\mathrm{Tr}_{|\Gamma_1} \eta = \mathrm{Tr}_{|\Gamma_1} z_1 + \mathrm{Tr}_{|\Gamma_1} P_{\alpha K}(\eta - (z_1 + u_2))$ *or equivalently*

$$\eta = (\mathrm{Tr}_{|\Gamma_1})^* \, \mathrm{Tr}_{|\Gamma_1} \left(z_1 + P_{\alpha K}(\eta - (z_1 + u_2))\right). \tag{4.12}$$

The proof follows analogue arguments as the one of Theorem 3.1.4 by just correctly replacing the projection π_{V_2} by the trace operator $\mathrm{Tr}_{|\Gamma_1}$ and by replacing the spaces V_i with the new ones respectively.

Analogue to Proposition 3.1.6 we can show the following result:

Proposition 4.2.3. *The following statements are equivalent:*

(i) there exists $\eta \in V_1$ *such that* $\eta = (\mathrm{Tr}_{|\Gamma_1})^* \, \mathrm{Tr}_{|\Gamma_1} (z_1 + P_{\alpha K}(\eta - (z_1 + u_2)))$ *(which is in turn the condition (iv) of Theorem 4.2.2)*

(ii) the sequence $(\eta^{(m)})_m$ *produced by the following iterative algorithm*

$$\begin{aligned} &\eta^{(0)} \in V_1, \ \mathrm{supp}\, \eta^{(0)} = \Gamma_1 \\ &\eta^{(m+1)} = (\mathrm{Tr}_{|\Gamma_1})^* \, \mathrm{Tr}_{|\Gamma_1} \left(z_1 + P_{\alpha K}(\eta^{(m)} - (z_1 + u_2))\right), \quad m \geq 0. \end{aligned} \tag{4.13}$$

converges to any $\eta \in V_1$ *that satisfies (4.12).*

Convergence of the subspace minimization

From the results of the previous section it follows that the iteration (4.7) can be explicitly computed by

$$u_1^{(\ell+1)} = \mathbb{S}_\alpha(u_1^{(\ell)} + T^*(g - Tu_2 - Tu_1^{(\ell)}) + u_2 - \eta^{(\ell)}) - u_2, \tag{4.14}$$

where $\mathbb{S}_\alpha := I - P_{\alpha K}$ and $\eta^{(\ell)} \in V_1$ is any solution of the fixed point equation

$$\begin{aligned} \eta = (\mathrm{Tr}_{|\Gamma_1})^* \, \mathrm{Tr}_{|\Gamma_1} \Big(&(u_1^{(\ell)} + T^*(g - Tu_2 - Tu_1^{(\ell)})) \\ &- P_{\alpha K}(u_1^{(\ell)} + T^*(g - Tu_2 - Tu_1^{(\ell)} + u_2 - \eta)) \Big). \end{aligned}$$

The computation of $\eta^{(\ell)}$ can be implemented by the algorithm in (4.13).

Proposition 4.2.4. *Assume* $u_2 \in V_2$ *and* $\|T\| < 1$. *Then the iteration* (4.14) *converges to a solution* $u_1^* \in V_1$ *of* (4.6) *for any initial choice of* $u_1^{(0)} \in V_1$.

The proof of this statement is analogue to the one of Theorem 3.1.9.

We conclude this section by mentioning that for the minimization on V_2 all the results presented here hold symmetrically by just adjusting the notations accordingly.

4.2.2 Convergence of the Sequential Domain Decomposition Method

In this section we want to prove the convergence of the algorithm in (4.5) to minimizers of $\mathcal{J}$. In order to do that, we need a characterization of solutions of the minimization problem (4.4) as the one provided in [113, Proposition 4.1] for the continuous setting and specified for the discrete setting in Proposition 3.2.2.

Convergence properties

We return to the sequential algorithm in (4.5). Let us explicitly express the algorithm as follows: pick an initial $V_1 + V_2 \ni \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} := u^{(0)} \in \mathcal{H}$, for example, $\tilde{u}_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = \tilde{u}_1^{(n)} \\ u_1^{(n+1,\ell+1)} = \arg\min_{\substack{u_1 \in V_1 \\ \mathrm{Tr}_{|\Gamma_1} u_1 = 0}} \mathcal{J}_1^s(u_1 + \tilde{u}_2^{(n)}, u_1^{(n+1,\ell)}) \quad \ell = 0, \ldots, L-1 \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = \tilde{u}_2^{(n)} \\ u_2^{(n+1,m+1)} = \arg\min_{\substack{u_2 \in V_2 \\ \mathrm{Tr}_{|\Gamma_2} u_2 = 0}} \mathcal{J}_2^s(u_1^{(n+1,L)} + u_2, u_2^{(n+1,m)}) \quad m = 0, \ldots, M-1 \end{cases} \\ u^{(n+1)} := u_1^{(n+1,L)} + u_2^{(n+1,M)} \\ \tilde{u}_1^{(n+1)} := \chi_1 \cdot u^{(n+1)} \\ \tilde{u}_2^{(n+1)} := \chi_2 \cdot u^{(n+1)}. \end{cases} \tag{4.15}$$

The algorithm in (4.15) consists of two nested iterations. The inner iterations with indexes ℓ and m constitute the iterative solution for the sequence of surrogate functionals on each subspace. Hence, these iterations approximatively compute minimizers for the functional $\mathcal{J}$ on the subspaces. The outer iteration with index n stems from our domain decomposition approach and iteratively computes the minimizer of $\mathcal{J}$ on the whole space. Note that we do prescribe a finite number L and M of inner iterations for each subspace respectively and that $u^{(n+1)} = \tilde{u}_1^{(n+1)} + \tilde{u}_2^{(n+1)}$, with $u_i^{(n+1)} \neq \tilde{u}_i^{(n+1)}$, $i = 1, 2$, in general. In this section we want to prove the convergence of the algorithm in (4.15) for any choice of L and M.

Proposition 4.2.5 (Convergence properties)**.** *Let us assume that $\|T\| < 1$. The algorithm in (4.15) produces a sequence $(u^{(n)})_n$ in $\mathcal{H}$ with the following properties:*

(i) $\mathcal{J}(u^{(n)}) > \mathcal{J}(u^{(n+1)})$ *for all* $n \in \mathbb{N}$ *(unless* $u^{(n)} = u^{(n+1)}$*);*

(ii) $\lim_{n\to\infty} \|u^{(n+1)} - u^{(n)}\|_2 = 0$*;*

(iii) the sequence $(u^{(n)})_n$ *has subsequences that converge in* $\mathcal{H}$*.*

We will skip the proof of this proposition, since it follows analogue arguments as the one of Theorem 3.1.12.

The use of the partition of unity $\{\chi_1, \chi_2\}$ allows not only to guarantee the boundedness of $(u^{(n)})_n$, but also of the sequences $(\tilde{u}_1^{(n)})_n$ and $(\tilde{u}_2^{(n)})_n$.

Lemma 4.2.6. *The sequences* $(\tilde{u}_1^{(n)})_n$ *and* $(\tilde{u}_2^{(n)})_n$ *produced by the algorithm in* (4.15) *are bounded, i.e., there exists a constant* $\tilde{C} > 0$ *such that* $\|\tilde{u}_i^{(n)}\|_2 \le \tilde{C}$ *for* $i = 1, 2$*.*

Proof. From the boundedness of $(u^{(n)})_n$ we have

$$\|\tilde{u}_i^{(n)}\|_2 = \|\chi_i u^{(n)}\|_2 \le c_\chi \|u^{(n)}\|_2 \le \tilde{C} \quad \text{for } i = 1, 2.$$

□

From Remark 3.2.3 (iii) we can also show the following auxiliary lemma.

Lemma 4.2.7. *The sequences* $(\eta_1^{(n,L)})_n$ *and* $(\eta_2^{(n,M)})_n$ *are bounded.*

Proof. From previous considerations we know that

$$\begin{aligned} u_1^{(n,L)} &= \mathbb{S}_\alpha(z_1^{(n,L-1)} + \tilde{u}_2^{(n-1)} - \eta_1^{(n,L)}) - \tilde{u}_2^{(n-1)} \\ u_2^{(n,M)} &= \mathbb{S}_\alpha(z_2^{(n,M-1)} + u_1^{(n,L)} - \eta_2^{(n,M)}) - u_1^{(n,L)}. \end{aligned}$$

Assume $(\eta_1^{(n,L)})_n$ were unbounded, then by Remark 3.2.3 (iii), also $\mathbb{S}_\alpha(z_1^{(n,L-1)} + \tilde{u}_2^{(n-1)} - \eta_1^{(n,L)})$ would be unbounded. By the monotonicity property of $\mathcal{J}$, see Proposition 4.2.5, and by (3.12) we obtain by analogue arguments as in the proof of Theorem 3.1.12 that

$$\left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_2^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_2^2 \right) \to 0, \quad n \to \infty. \tag{4.16}$$

Since $(\tilde{u}_2^{(n)})_n$ and $(u_1^{(n,L)})_n$ are bounded by Lemma 4.2.6 and formula (4.16), we have a contradiction. Thus $(\eta_1^{(n,L)})_n$ has to be bounded. With the same argument we can show that $(\eta_2^{(n,M)})_n$ is bounded. □

Convergence to Minimizers

Now we are eventually able to show that the algorithm in (4.15) is indeed converging to a minimizer of the original functional $\mathcal{J}$.

Theorem 4.2.8 (Convergence to minimizers)**.** *Assume* $\|T\| < 1$*. Then accumulation points of the sequence* $(u^{(n)})_n$ *produced by the algorithm in (4.15) are minimizers of* $\mathcal{J}$*. If* $\mathcal{J}$ *has a unique minimizer, then the sequence* $(u^{(n)})_n$ *converges to it.*

Proof. Let us denote $u^{(\infty)}$ the limit of a subsequence. For simplicity, we rename such a subsequence by $(u^{(n)})_n$. From Lemma 4.2.6 we know that $(\tilde{u}_1^{(n)})_n$, $(\tilde{u}_2^{(n)})_n$ and consequently $(u_1^{(n,L)})_n$,$(u_2^{(n,M)})_n$ are bounded. So the limit $u^{(\infty)}$ can be written as

$$u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)} = \tilde{u}_1^{(\infty)} + \tilde{u}_2^{(\infty)}, \tag{4.17}$$

where $u_1^{(\infty)}$ is the limit of $(u_1^{(n,L)})_n$, $u_2^{(\infty)}$ is the limit of $(u_2^{(n,M)})_n$, and $\tilde{u}_i^{(\infty)}$ is the limit of $(\tilde{u}_i^{(n)})_n$ for $i = 1, 2$. Now we show that $\tilde{u}_2^{(\infty)} = u_2^{(\infty)}$. By using the triangle inequality, from (4.16) it directly follows that

$$\|u_2^{(n+1,M)} - \tilde{u}_2^{(n)}\|_2 \to 0, \quad n \to \infty. \tag{4.18}$$

Moreover, since $\chi_2 \in V_2$ is a fixed vector which is independent of n, we obtain from Proposition 4.2.5 (ii) that

$$\|\chi_2(u^{(n)} - u^{(n+1)})\|_2 \to 0, \quad n \to \infty,$$

and hence

$$\|\tilde{u}_2^{(n)} - \tilde{u}_2^{(n+1)}\|_2 \to 0, \quad n \to \infty. \tag{4.19}$$

Putting (4.18) and (4.19) together and noting that

$$\|u_2^{(n+1,M)} - \tilde{u}_2^{(n)}\|_2 + \|\tilde{u}_2^{(n)} - \tilde{u}_2^{(n+1)}\|_2 \geq \|u_2^{(n+1,M)} - \tilde{u}_2^{(n+1)}\|_2$$

we have

$$\|u_2^{(n+1,M)} - \tilde{u}_2^{(n+1)}\|_2 \to 0, \quad n \to \infty, \tag{4.20}$$

which means that the sequences $(u_2^{(n,M)})_n$ and $(\tilde{u}_2^{(n)})_n$ have the same limit, i.e., $\tilde{u}_2^{(\infty)} = u_2^{(\infty)}$, which we denote by $u_2^{(\infty)}$. Then from (4.20) and (4.17) it directly follows that $\tilde{u}_1^{(\infty)} = u_1^{(\infty)}$.

We set

$$F_1(u_1^{(n+1,L)}) := \|u_1^{(n+1,L)} - z_1^{(n+1,L)}\|_2^2 + 2\alpha|\nabla(u_1^{(n+1,L)} + \tilde{u}^{(n)}_{2|_{\Omega_1}})|(\Omega_1),$$

where

$$z_1^{(n+1,L)} := u_1^{(n+1,L-1)} + (T^*(g - T\tilde{u}_2^{(n)} - Tu_1^{(n+1,L-1)}))_{|_{\Omega_1}}.$$

The optimality condition for $u_1^{(n+1,L)}$ is

$$0 \in \partial_{V_1} F_1(u_1^{(n+1,L)}) + 2\eta_1^{(n+1,L)},$$

where

$$\eta_1^{(n+1,L)} = (\mathrm{Tr}_{|\Gamma_1})^* \,\mathrm{Tr}_{|\Gamma_1} \left((z_1^{(n+1,L)}) + P_{\alpha K}(\eta_1^{(n+1,L)} - z_1^{(n+1,L)} - \tilde{u}_2^{(n)}) \right).$$

In order to use the characterization of elements in the subdifferential of $|\nabla u|(\Omega)$, i.e., Proposition 3.2.2, we have to rewrite the minimization problem for F_1. More precisely, we define

$$\hat{F}_1(\xi_1^{(n+1,L)}) := \|\xi_1^{(n+1,L)} - \tilde{u}^{(n)}_{2|\Omega_1} - z_1^{(n+1,L)}\|_2^2 + 2\alpha|\nabla(\xi_1^{(n+1,L)})|(\Omega_1)$$

for $\xi_1^{(n+1,L)} \in V_1$ with $\mathrm{Tr}_{|\Gamma_1} \xi_1^{(n+1,L)} = \tilde{u}_2^{(n)}$. Then the optimality condition for $\xi_1^{(n+1,L)}$ is

$$0 \in \partial \hat{F}_1(\xi_1^{(n+1,L)}) + 2\eta_1^{(n+1,L)}. \tag{4.21}$$

Note that indeed $\xi_1^{(n+1,L)}$ is optimal if and only if $u_1^{(n+1,L)} = \xi_1^{(n+1,L)} - \tilde{u}^{(n)}_{2|\Omega_1}$ is optimal.

Analogously we define

$$\hat{F}_2(\xi_2^{(n+1,M)}) := \|\xi_2^{(n+1,M)} - u^{(n+1,L)}_{1|\Omega_2} - z_2^{(n+1,M)}\|_2^2 + 2\alpha|\nabla(\xi_2^{(n+1,M)})|(\Omega_2)$$

for $\xi_2^{(n+1,M)} \in V_2$ with $\mathrm{Tr}_{|\Gamma_2} \xi_2^{(n+1,M)} = u_1^{(n+1,L)}$, and the optimality condition for $\xi_2^{(n+1,M)}$ is

$$0 \in \partial \hat{F}_2(\xi_2^{(n+1,M)}) + 2\eta_2^{(n+1,M)}, \tag{4.22}$$

where

$$\eta_2^{(n+1,M)} = (\mathrm{Tr}_{|\Gamma_2})^* \,\mathrm{Tr}_{|\Gamma_2} \left((z_2^{(n+1,M)}) + P_{\alpha K}(\eta_2^{(n+1,M)} - z_2^{(n+1,M)} - u_1^{(n+1,L)}) \right).$$

Let us recall that now we are considering functionals as in Proposition 3.2.2 with $\varphi(s) = s$, $T = I$, and $\Omega = \Omega_i$, $i = 1, 2$. From Proposition 3.2.2 and Remark 3.2.3 we get that $\xi_1^{(n+1,L)}$, and consequently $u_1^{(n+1,L)}$ is optimal, i.e., $-2\eta_1^{(n+1,L)} \in \partial \hat{F}_1(\xi_1^{(n+1,L)})$, if and only if there exists an $M_1^{(n+1)} = (M_{0,1}^{(n+1)}, \bar{M}_1^{(n+1)}) \in V_1 \times V_1^d$ with $|\bar{M}_1^{(n+1)}(x)| \leq 2\alpha$ for all $x \in \Omega_1$ such that

$$\langle \bar{M}_1^{(n+1)}(x), (\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}))(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}))(x)| = 0 \tag{4.23}$$

$$-2(u_1^{(n+1,L)}(x) - z_1^{(n+1,L)}(x)) - \operatorname{div} \bar{M}_1^{(n+1)}(x) - 2\eta_1^{(n+1,L)}(x) = 0, \tag{4.24}$$

for all $x \in \Omega_1$. Analogously we get that $\xi_2^{(n+1,M)}$, and consequently $u_2^{(n+1,M)}$ is optimal, i.e., $-2\eta_2^{(n+1,M)} \in \partial \hat{F}_2(\xi_2^{(n+1,M)})$, if and only if there exists an $M_2^{(n+1)} = (M_{0,2}^{(n+1)}, \bar{M}_2^{(n+1)}) \in V_2 \times V_2^d$ with $|\bar{M}_2^{(n+1)}(x)| \leq 2\alpha$ for all $x \in \Omega_2$ such that

$$\langle \bar{M}_2^{(n+1)}(x), (\nabla(u_1^{(n+1,L)} + u_2^{(n+1,M)}))(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n+1,M)}))(x)| = 0 \tag{4.25}$$

$$-2(u_2^{(n+1,M)}(x) - z_2^{(n+1,M)}(x)) - \operatorname{div} \bar{M}_2^{(n+1)}(x) - 2\eta_2^{(n+1,M)}(x) = 0, \tag{4.26}$$

for all $x \in \Omega_2$. Since $(\bar{M}_1^{(n)}(x))_n$ is bounded for all $x \in \Omega_1$ and $(\bar{M}_2^{(n)}(x))_n$ is bounded for all $x \in \Omega_2$, there exist convergent subsequences $(\bar{M}_1^{(n_k)}(x))_k$ and $(\bar{M}_2^{(n_k)}(x))_k$. Let us denote $\bar{M}_1^{(\infty)}(x)$ and $\bar{M}_2^{(\infty)}(x)$ the respective limits of the sequences. For simplicity we rename such sequences by $(\bar{M}_1^{(n)}(x))_n$ and $(\bar{M}_2^{(n)}(x))_n$.

Note that, by Lemma 4.2.7 (or simply from (4.24) and (4.26)) the sequences $(\eta_1^{(n,L)})_n$ and $(\eta_2^{(n,M)})_n$ are also bounded. Hence there exist convergent subsequences that we denote, for simplicity, again by $(\eta_1^{(n,L)})_n$ and $(\eta_2^{(n,M)})_n$ with limits $\eta_i^{(\infty)}$, $i = 1, 2$. By taking in (4.23)-(4.26) the limits for $n \to \infty$ we obtain

$$\begin{aligned} \langle \bar{M}_1^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| = 0 \quad & \text{for all } x \in \Omega_1 \\ -2(u_1^{(\infty)}(x) - z_1^{(\infty)}(x)) - \operatorname{div} \bar{M}_1^{(\infty)}(x) - 2\eta_1^{(\infty)}(x) = 0 \quad & \text{for all } x \in \Omega_1 \end{aligned}$$

$$\begin{aligned} \langle \bar{M}_2^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| = 0 \quad & \text{for all } x \in \Omega_2 \\ -2(u_2^{(\infty)}(x) - z_2^{(\infty)}(x)) - \operatorname{div} \bar{M}_2^{(\infty)}(x) - 2\eta_2^{(\infty)}(x) = 0 \quad & \text{for all } x \in \Omega_2. \end{aligned}$$

Since $\operatorname{supp} \eta_1^{(\infty)} = \Gamma_1$ and $\operatorname{supp} \eta_2^{(\infty)} = \Gamma_2$ we have

$$\begin{aligned} \langle \bar{M}_1^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| = 0 \quad & \text{for all } x \in \Omega_1 \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \bar{M}_1^{(\infty)}(x) = 0 \quad & \text{for all } x \in \Omega_1 \setminus \Gamma_1 \end{aligned} \tag{4.27}$$

$$\begin{aligned} \langle \bar{M}_2^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| = 0 \quad & \text{for all } x \in \Omega_2 \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \bar{M}_2^{(\infty)}(x) = 0 \quad & \text{for all } x \in \Omega_2 \setminus \Gamma_2. \end{aligned} \tag{4.28}$$

Observe now that from Proposition 3.2.2 we also have that $0 \in \mathcal{J}(u^{(\infty)})$ if and only if there exists $M^{(\infty)} = (M_0^{(\infty)}, \bar{M}^{(\infty)})$ with $|\bar{M}^{(\infty)}(x)| \leq 2\alpha$ for all $x \in \Omega$ such that

$$\begin{aligned} \langle \bar{M}^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| = 0 \quad & \text{for all } x \in \Omega \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \bar{M}^{(\infty)}(x) = 0 \quad & \text{for all } x \in \Omega. \end{aligned} \tag{4.29}$$

Note that $\bar{M}_j^{(\infty)}(x), j = 1, 2$, for $x \in \Omega_1 \cap \Omega_2$ satisfies both (4.27) and (4.28). Hence let us choose

$$M^{(\infty)}(x) = \begin{cases} M_1^{(\infty)}(x) & \text{if } x \in \Omega_1 \setminus \Gamma_1 \\ M_2^{(\infty)}(x) & \text{if } x \in (\Omega_2 \setminus \Omega_1) \cup \Gamma_1 \end{cases}.$$

With this choice of $M^{(\infty)}$ the equations (4.27) - (4.29) are valid and hence $u^{(\infty)}$ is optimal in Ω. □

Remark 4.2.9. *(i) If $\nabla u^{(\infty)}(x) \neq 0$ for $x \in \Omega_j$, $j = 1, 2$, then $\bar{M}_j^{(\infty)}$ is given as in equation* (3.37) *by*

$$\bar{M}_j^{(\infty)}(x) = -2\alpha \frac{(\nabla u^{(\infty)}_{|\Omega_j})(x)}{|(\nabla u^{(\infty)}_{|\Omega_j})(x)|}.$$

(ii) The boundedness of the sequences $(\tilde{u}_1^{(n)})_n$ and $(\tilde{u}_2^{(n)})_n$ has been technically used for showing the existence of an optimal decomposition $u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)}$ in the proof of Theorem 4.2.8. Their boundedness is guaranteed as in Lemma 4.2.6 by the use of the partition of the unity $\{\chi_1, \chi_2\}$. Let us emphasize that there is no way of obtaining the boundedness of the local sequences $(u_1^{(n,L)})_n$ and $(u_2^{(n,M)})_n$ otherwise. In Figure 4.6 we show that the local sequences can become unbounded in case we do not modify them by means of the partition of the unity.

(iii) Note that for deriving the optimality condition (4.29) *for $u^{(\infty)}$ we combined the respective conditions* (4.27) *and* (4.28) *for $u_1^{(\infty)}$ and $u_2^{(\infty)}$. In doing that, we strongly took advantage of the overlapping property of the subdomains, hence avoiding a fine analysis of $\eta_1^{(\infty)}$ and $\eta_2^{(\infty)}$ on the interfaces Γ_1 and Γ_2.*

Remark 4.2.10. *The generalization of the algorithm to a multiple domain decomposition is straightforward. Let us split now Ω into $\mathcal{N} \geq 2$ overlapping domains Ω_i, $i = 1, \ldots, \mathcal{N}$. Associated with this decomposition we define $V_i := \{u \in \mathcal{H} : \operatorname{supp}(u) \subset \Omega_i\}$ such that $\mathcal{H} = V_1 + \ldots + V_{\mathcal{N}}$ and we denote $u_i = \pi_{V_i} u$ for $i = 1, \ldots, \mathcal{N}$. By $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$ we denote the inner interfaces of the domain patches. Further we fix a bounded uniform partition of unity (BUPU) $\{\chi_1, \ldots, \chi_{\mathcal{N}}\} \subset \mathcal{H}$ such that*

(a) $\mathrm{Tr}_{|\Gamma_i} \chi_i = 0$ *for* $i = 1, \ldots, \mathcal{N}$,

(b) $\sum_{i=1}^{\mathcal{N}} \chi_i = 1$,

(c) $\operatorname{supp} \chi_i \subset \Omega_i$ *for* $i = 1, \ldots, \mathcal{N}$,

(d) $\max\{\|\chi_1\|_\infty, \ldots, \|\chi_{\mathcal{N}}\|_\infty\} = c_\chi < \infty$.

Then we define the overlapping multiple domain decomposition algorithm as follows: pick an initial $V_1 + \ldots + V_{\mathcal{N}} \ni \tilde{u}_1^{(0)} + \ldots + \tilde{u}_{\mathcal{N}}^{(0)} := u^{(0)} \in \mathcal{H}$, for example, $\tilde{u}_i^{(0)} = 0$,

$i = 1, \dots, \mathcal{N}$, and iterate

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} u_1^{(n+1,0)} = \tilde{u}_1^{(n)} \\ u_1^{(n+1,\ell+1)} = \arg\min_{\substack{u_1 \in V_1 \\ \mathrm{Tr}|_{\Gamma_1} u_1 = 0}} \mathcal{J}_1^s(u_1 + \sum_{i=2}^{\mathcal{N}} \tilde{u}_i^{(n)}, u_1^{(n+1,\ell)}) \\ \ell = 0, \dots, L_1 - 1 \end{array} \right. \\ \dots \\ \left\{ \begin{array}{l} u_{\mathcal{N}}^{(n+1,0)} = \tilde{u}_{\mathcal{N}}^{(n)} \\ u_{\mathcal{N}}^{(n+1,\ell+1)} = \arg\min_{\substack{u_{\mathcal{N}} \in V_{\mathcal{N}} \\ \mathrm{Tr}|_{\Gamma_{\mathcal{N}}} u_{\mathcal{N}} = 0}} \mathcal{J}_{\mathcal{N}}^s(\sum_{i=1}^{\mathcal{N}-1} u_i^{(n+1,L)} + u_{\mathcal{N}}, u_{\mathcal{N}}^{(n+1,\ell)}) \\ \ell = 0, \dots, L_{\mathcal{N}} - 1 \end{array} \right. \\ u^{(n+1)} := \sum_{i=1}^{\mathcal{N}} u_i^{(n+1,L)} \\ \tilde{u}_i^{(n+1)} := \chi_i \cdot u^{(n+1)} \quad \textit{for } i = 1, \dots, \mathcal{N}. \end{array} \right. \tag{4.30}$$

The surrogate functionals $\mathcal{J}_i^s$ are defined in an analogous way as above, for instance, $\mathcal{J}_1^s$ is given as in (4.8) *by just substituting $\sum_{i=2}^{\mathcal{N}} u_i^{(n)}$ for u_2 and by using the appropriate spaces. Then one can show the same convergence properties as in Theorem 4.2.5 and Theorem 4.2.8. Hence the convergence of algorithm in* (4.30) *to a minimizer of the original functional* (4.1) *is ensured.*

4.2.3 A Parallel Algorithm and its Convergence

One of the main motivations and success of domain decomposition methods is the reduction of the dimension with parallelization. In this subsection we introduce a possible parallelization of the previous algorithm in (4.15), which may read as follows: pick an initial $V_1 + V_2 \ni \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} := u^{(0)} \in \mathcal{H}$, for example, $\tilde{u}_i^{(0)} = 0$, $i = 1, 2$, and iterate

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} u_1^{(n+1,0)} = \tilde{u}_1^{(n)} \\ u_1^{(n+1,\ell+1)} = \arg\min_{\substack{u_1 \in V_1 \\ \mathrm{Tr}|_{\Gamma_1} u_1 = 0}} \mathcal{J}_1^s(u_1 + \tilde{u}_2^{(n)}, u_1^{(n+1,\ell)}) \quad \ell = 0, \dots, L-1 \end{array} \right. \\ \left\{ \begin{array}{l} u_2^{(n+1,0)} = \tilde{u}_2^{(n)} \\ u_2^{(n+1,m+1)} = \arg\min_{\substack{u_2 \in V_2 \\ \mathrm{Tr}|_{\Gamma_2} u_2 = 0}} \mathcal{J}_2^s(\tilde{u}_1^{(n)} + u_2, u_2^{(n+1,m)}) \quad m = 0, \dots, M-1 \end{array} \right. \\ u^{(n+1)} := \frac{u_1^{(n+1,L)} + u_2^{(n+1,M)} + u^{(n)}}{2} \\ \tilde{u}_1^{(n+1)} := \chi_1 \cdot u^{(n+1)} \\ \tilde{u}_2^{(n+1)} := \chi_2 \cdot u^{(n+1)}. \end{array} \right. \tag{4.31}$$

We are going to propose similar convergence results as for the sequential algorithm.

Proposition 4.2.11 (Convergence properties)**.** *Let us assume that $\|T\| < 1$. The parallel algorithm in (4.31) produces a sequence $(u^{(n)})_n$ in $\mathcal{H}$ with the following properties:*

(i) $\mathcal{J}(u^{(n)}) > \mathcal{J}(u^{(n+1)})$ *for all* $n \in \mathbb{N}$ *(unless* $u^{(n)} = u^{(n+1)}$*);*

(ii) $\lim_{n\to\infty} \|u^{(n+1)} - u^{(n)}\|_2 = 0$*;*

(iii) the sequence $(u^{(n)})_n$ *has subsequences that converge in* $\mathcal{H}$*.*

Proof. By the same argument as in the proof of Theorem 3.1.12, we obtain

$$\mathcal{J}(u^{(n)}) - \mathcal{J}(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}) \geq C \sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_2^2$$

and

$$\mathcal{J}(u^{(n)}) - \mathcal{J}(\tilde{u}_1^{(n)} + u_2^{(n+1,M)}) \geq C \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_2^2.$$

Hence, by summing and halving

$$\begin{aligned} & \mathcal{J}(u^{(n)}) - \frac{1}{2}(\mathcal{J}(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}) + \mathcal{J}(\tilde{u}_1^{(n)} + u_2^{(n+1,M)})) \\ \geq \; & \frac{C}{2}\left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_2^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_2^2\right). \end{aligned}$$

We recall that $\mathcal{J}(u^{(n)}) = \|Tu^{(n)} - g\|_2^2 + 2\alpha|\nabla u^{(n)}|(\Omega)$ (and T is linear). Then, by the standard inequality $(a^2+b^2) \geq \frac{1}{2}(a+b)^2$ for $a, b > 0$, we have

$$\begin{aligned} \left\|Tu^{(n+1)} - g\right\|_2^2 &= \left\|T\left(\frac{(u_1^{(n+1,L)} + u_2^{(n+1,M)}) + u^{(n)}}{2}\right) - g\right\|_2^2 \\ &\leq \frac{1}{2}\|T(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}) - g\|_2^2 + \frac{1}{2}\|T(\tilde{u}_1^{(n)} + u_2^{(n+1,M)}) - g\|_2^2. \end{aligned}$$

Moreover we have

$$|\nabla(u^{(n+1)})|(\Omega) \;\leq\; \frac{1}{2}\left(|\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n)})|(\Omega) + |\nabla(\tilde{u}_1^{(n)} + u_2^{(n+1,M)})|(\Omega)\right).$$

By the last two inequalities we immediately show that

$$\mathcal{J}(u^{(n+1)}) \leq \frac{1}{2}\left(\mathcal{J}(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}) + \mathcal{J}(\tilde{u}_1^{(n)} + u_2^{(n+1,M)})\right),$$

hence

$$\begin{aligned} & \mathcal{J}(u^{(n)}) - \mathcal{J}(u^{(n+1)}) \\ \geq \; & \frac{C}{2}\left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_2^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_2^2\right) \geq 0. \end{aligned} \tag{4.32}$$

Since the sequence $(\mathcal{J}(u^{(n)}))_n$ is monotonically decreasing and bounded from below by 0, it is also convergent. From (4.32) and the latter convergence we deduce

$$\left(\sum_{\ell=0}^{L-1} \|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\|_2^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\|_2^2 \right) \to 0, \quad n \to \infty. \tag{4.33}$$

In particular, by again using $(a^2+b^2) \geq \frac{1}{2}(a+b)^2$ for $a, b > 0$ and the triangle inequality, we also have

$$\|u^{(n)} - u^{(n+1)}\|_2 \to 0, \quad n \to \infty. \tag{4.34}$$

The rest of the proof follows by analogous arguments as in that of Theorem 3.1.12. □

Analogous results as the one stated in Lemma 4.2.6 and Lemma 4.2.7 also hold in the parallel case. With these preliminary results the following theorem holds:

Theorem 4.2.12 (Convergence to minimizers)**.** *Assume $\|T\| < 1$. Then accumulation points of the sequence $(u^{(n)})_n$ produced by algorithm (4.31) are minimizers of $\mathcal{J}$. If $\mathcal{J}$ has a unique minimizer, then the sequence $(u^{(n)})_n$ converges to it.*

Proof. Note that $u^{(n+1)}$ is the average of the current iteration and the previous, i.e.,

$$u^{(n+1)} = \frac{u_1^{(n+1,L)} + u_2^{(n+1,M)} + u^{(n)}}{2}.$$

Observe that the sequences $(u_1^{(n+1,L)})_n$, $(u_2^{(n+1,M)})_n$ and $(u^{(n)})_n$ are bounded. Hence there exist convergent subsequences. By taking the limit for $n \to \infty$ we obtain

$$u^{(\infty)} = \frac{u_1^{(\infty)} + u_2^{(\infty)} + u^{(\infty)}}{2},$$

which is equivalent to

$$u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)}.$$

With this observation the rest of the proof follows by analogous arguments as in that of Theorem 4.2.8. □

4.2.4 Applications and Numerics for the Sequential Implementation

In this section we shall present the application of the sequential and parallel algorithms (4.15) and (4.31) for the minimization of $\mathcal{J}$ in one and two dimensions. In particular, we give a detailed explanation of the domain decompositions used in the numerics. Furthermore we present numerical examples for image inpainting, i.e., the recovery of missing parts of images by minimal total variation interpolation, and for compressed sensing [16, 17, 18, 52], more specifically, the nonadaptive compressed acquisition of images for a classical toy problem inspired by magnetic resonance imaging (MRI) [17, 81]. The numerical examples of this section and respective Matlab codes can be found at [126]. These Matlab codes represent the implementation of our proposed methods. Additionally, all the examples included in this chapter are produced with these codes.

Domain Decomposition

In one dimension the domain Ω is a set of N equidistant points on an interval $[a, b]$, i.e., $\Omega = \{a = x_1, \ldots, x_N = b\}$ and is split into two overlapping intervals Ω_1 and Ω_2. Let $|\Omega_1 \cap \Omega_2| =: G$ be the size of the overlap of Ω_1 and Ω_2. Then we set $\Omega_1 = \{a = x_1, \ldots, x_{n_1}\}$ and $\Omega_2 = \{x_{n_1-G+1}, \ldots, x_N = b\}$ with $|\Omega_1| := n_1 = \left\lceil \frac{N+G}{2} \right\rceil$. The interfaces Γ_1 and Γ_2 are located in $i = n_1$ and $n_1 - G + 1$ respectively (cf. Figure 4.2). The auxiliary functions χ_1 and χ_2 can be chosen in the following way (cf. Figure 4.1):

$$\chi_1(x_i) = \begin{cases} 1 & x_i \in \Omega_1 \setminus \Omega_2 \\ 1 - \frac{1}{G-1}(i - (n_1 - G + 1)) & x_i \in \Omega_1 \cap \Omega_2 \end{cases}$$

$$\chi_2(x_i) = \begin{cases} 1 & x_i \in \Omega_2 \setminus \Omega_1 \\ \frac{1}{G-1}(i - (n_1 - G + 1)) & x_i \in \Omega_1 \cap \Omega_2 \end{cases}.$$

Note that $\chi_1(x_i) + \chi_2(x_i) = 1$ for all $x_i \in \Omega$ (i.e for all $i = 1, \ldots, N$).

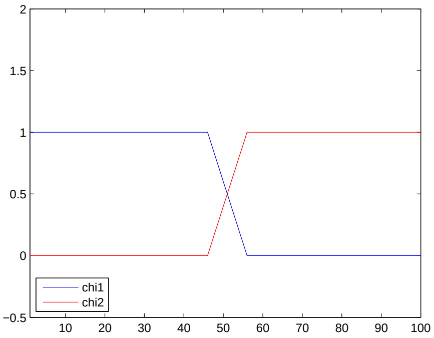


Figure 4.1: *Auxiliary functions χ_1 and χ_2 for an overlapping domain decomposition with two subdomains.*

In two dimensions the domain Ω, i.e., the set of $N_1 \times N_2$ equidistant points on the 2-dimensional rectangle $[a, b] \times [c, d]$, is split in an analogous way with respect to its rows. In particular we have Ω_1 and Ω_2 consisting of equidistant points on $[a, x_{n_1}] \times [c, d]$ and on $[x_{n_1-G+1}, b] \times [c, d]$ respectively, compare Figure 4.3. In more than two domains the splitting is done similarly.

Set $\Omega = \Omega_1 \cup \ldots \cup \Omega_{\mathcal{N}}$, the domain Ω decomposed into $\mathcal{N}$ domains Ω_i, $i = 1, \ldots, \mathcal{N}$, where Ω_i and Ω_{i+1} are overlapping for $i = 1, \ldots, \mathcal{N} - 1$. Let $|\Omega_i \cap \Omega_{i+1}| =: G$ for every $i = 1, \ldots, \mathcal{N} - 1$. Set $s = \lceil N_1/\mathcal{N} \rceil$. Then

```
$\Omega_1 = \left\{a = x_1, \ldots, x_{s+\frac{G}{2}}\right\} \times \{c = y_1, \ldots, y_{N_2} = d\}$
for $i = 2 : \mathcal{N} - 1$
    $\Omega_i = \left\{x_{(i-1)s-\frac{G}{2}+1}, \ldots, x_{is+\frac{G}{2}}\right\} \times \{c = y_1, \ldots, y_{N_2} = d\}$
end
$\Omega_{\mathcal{N}} = [x_{(\mathcal{N}-1)s-\frac{G}{2}+1}, x_{N_1}] \times \{c = y_1, \ldots, y_{N_2} = d\}\,.$
```

The auxiliary functions χ_i can be chosen in an analogous way as in the one dimensional case:

$$\chi_i(x_{i_1}, y_{i_2}) = \begin{cases} \frac{1}{G-1}(i_1 - ((i-1)s - G/2 + 1)) & (x_{i_1}, y_{i_2}) \in \Omega_{i-1} \cap \Omega_i \\ 1 & (x_{i_1}, y_{i_2}) \in \Omega_i \setminus (\Omega_{i-1} \cup \Omega_{i+1}) \\ 1 - \frac{1}{G-1}(i_1 - (is - G/2 + 1)) & (x_{i_1}, y_{i_2}) \in \Omega_i \cap \Omega_{i+1} \end{cases}$$

for $i = 1, \ldots, \mathcal{N}$ with $\Omega_0 = \Omega_{\mathcal{N}+1} = \emptyset$.

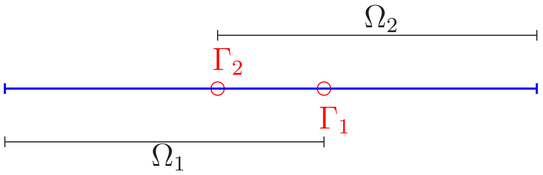


Figure 4.2: *Overlapping domain decomposition in 1D.*

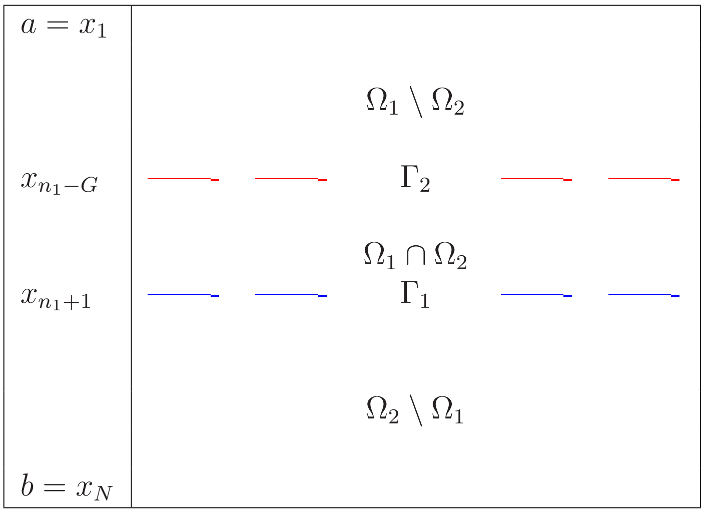


Figure 4.3: *Decomposition of the image in two domains* Ω_1 *and* Ω_2.

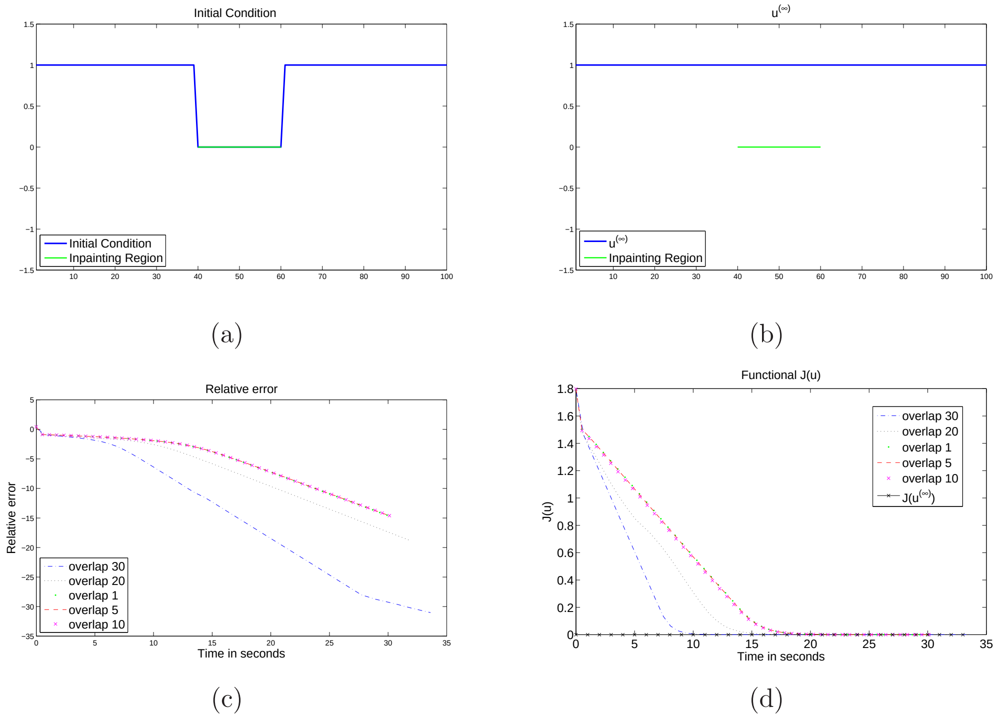
To compute the fixed point η of (4.12) in an efficient way we make the following considerations, which allow to restrict the computation from Ω_1 to a relatively small stripe around the interface. The fixed point η is actually supported on Γ_1 only, i.e., $\eta(x) = 0$ in $\Omega_1 \setminus \Gamma_1$. Hence, we restrict the fixed point iteration for η to a relatively small stripe $\hat{\Omega}_1 \subset \Omega_1$. Analogously, one implements the minimizations of η_2 on $\hat{\Omega}_2$. A similar trick was also used in [66] to compute suitable Lagrange multipliers at the interfaces of the non-overlapping domains. However, there it is needed to consider larger "bilateral stripes" around the support of the multiplier, making the numerical computation slightly more demanding for that algorithm.

Numerical Experiments for the Sequential Implementation

In the following we present numerical examples for the sequential algorithm in (4.15) in two particular applications: signal interpolation/image inpainting, and compressed sensing. The scope of the section is to illustrate by simple examples the main properties of the algorithms, as proven in our theoretical analysis. In particular, we emphasize the monotonicity properties of the algorithms with respect to the energy $\mathcal{J}$, the boundedness of the iterations due to the implementation of BUPUs, and the robustness in correctly computing minimizers independently of the size of overlapping regions.

In the numerical experiments the value for the parameter α has been chosen experimentally, i.e., we chose the value that gave the best compromise between visual quality of the minimizer and computational time of the algorithm. Note however, that there exist more systematic ways in order to choose an optimal value for α, where the choice depends both on the data noise level and the exact solution of the problem, cf., e.g., [55] for a general approach in regularized inverse problems, or [27] for a discussion of the correspondence between the noise level and α in the case of total variation minimization.

In Figure 4.4 and Figure 4.5 we show a partially corrupted 1D signal on an interval Ω of 100 sampling points, with a loss of information on an interval $D \subset \Omega$. The domain D of the missing signal points is marked with green. These signal points are reconstructed by total variation interpolation, i.e., minimizing the functional $\mathcal{J}$ in (4.1) with $\alpha = 0.4$ and $Tu = 1_{\Omega \setminus D} \cdot u$, where $1_{\Omega \setminus D}$ is the indicator function of $\Omega \setminus D$. A minimizer $u^{(\infty)}$ of $\mathcal{J}$ is precomputed with an algorithm working on the whole interval Ω without any decomposition. We show also the decay of relative error and of the value of the energy $\mathcal{J}$ for applications of algorithm in (4.15) on two subdomains and with different overlap sizes $G = 1, 5, 10, 20, 30$. The fixed points η's are computed on a small interval $\hat{\Omega}_i$, $i = 1, 2$, of size 2. These results confirm the behavior of the algorithm in (4.15) as predicted by the theory; the algorithm monotonically decreases $\mathcal{J}$ and computes a minimizer, independently of the size of the overlapping region. A larger overlapping region does not necessarily imply a slower convergence. In these figures we do compare the speed in terms of CPU time. In Figure 4.6 we also illustrate the effect of implementing the BUPU within the



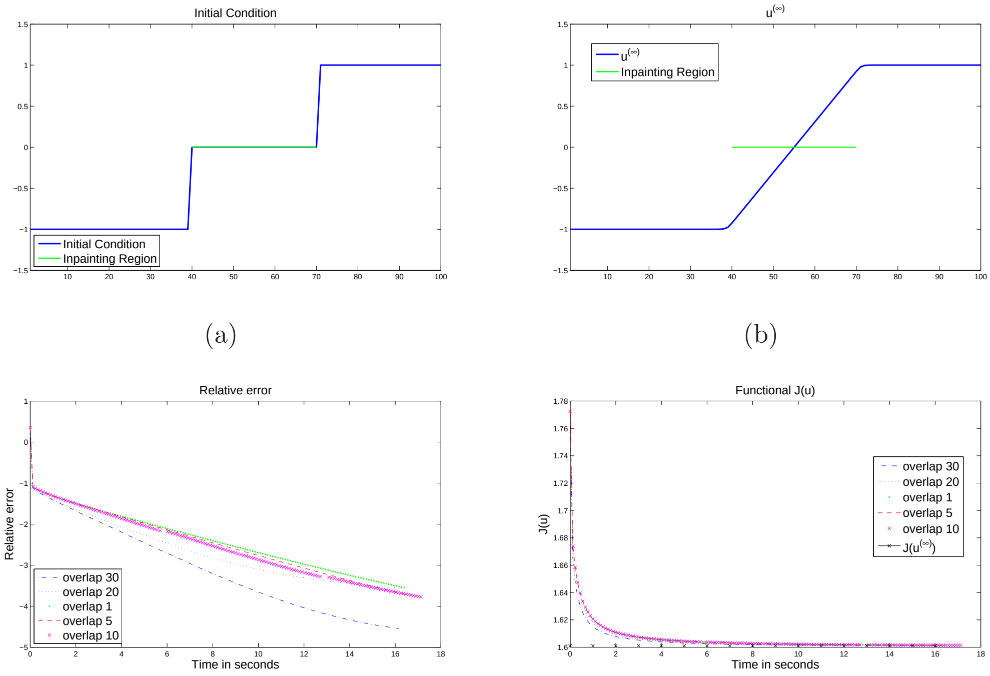
(a)

(b)

(c)

(d)

Figure 4.4: *We present a numerical experiment related to the interpolation of a 1D signal by total variation minimization. The original signal is only provided outside of the green subinterval. The initial datum g is shown in (a). As expected, the minimizer* $u^{(\infty)}$ *is the constant vector* 1*, as shown in (b). In (c) and (d) we display the rates of decay of the relative error and of the value of* $\mathcal{J}$ *respectively, for applications of the algorithm in* (4.15) *with different sizes G=1,5,10,20,30 of the overlapping region of two subintervals.*



(a)

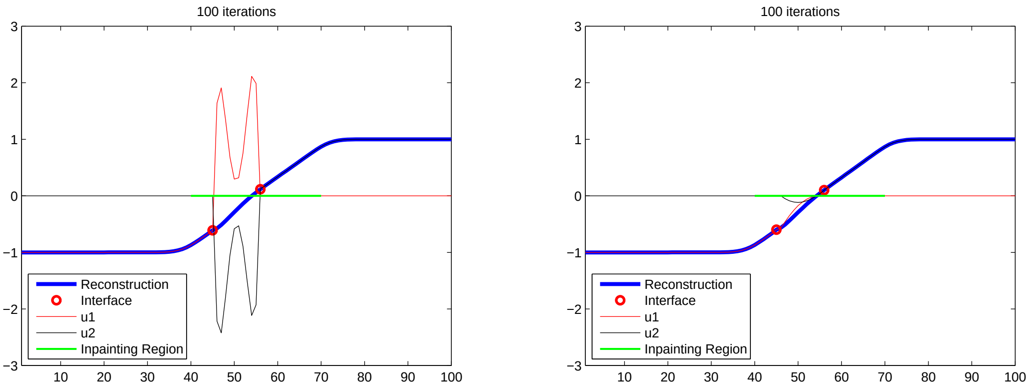
(b)

(c)

(d)

Figure 4.5: *We show a second example of total variation interpolation in 1D. The initial datum g is shown in (a). As expected, a minimizer $u^{(\infty)}$ is (nearly) a piecewise linear function, as shown in (b). In (c) and (d) we display the rates of decay of the relative error and of the value of $\mathcal{J}$ respectively, for applications of the algorithm in* (4.15) *with different sizes G=1,5,10,20,30 of the overlapping region of two subintervals.*

domain decomposition algorithm. In this case, with datum g as in Figure 4.5, we chose $\alpha = 1$ and an overlap of size $G = 10$. The fixed points η's are computed on a small interval $\hat{\Omega}_i$, $i = 1, 2$ respectively, of size 6.

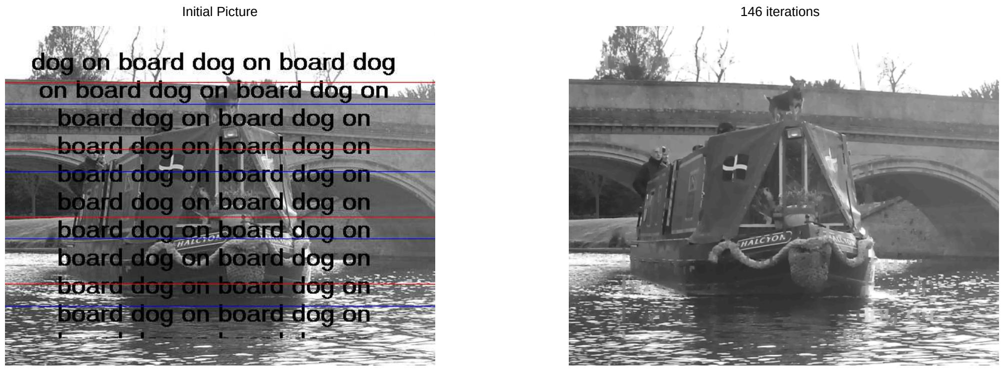


(a) (b)

Figure 4.6: *Here we present two numerical experiments related to the interpolation of a 1D signal by total variation minimization. The original signal is only provided outside of the green subinterval. On the left we show an application of the algorithm in* (4.15) *when no correction with the partition of unity is provided. In this case, the sequence of the local iterations* $u_1^{(n)}, u_2^{(n)}$ *is unbounded. On the right we show an application of the algorithm in* (4.15) *with the use of the partition of unity, which enforces the uniform boundedness of the local iterations* $u_1^{(n)}, u_2^{(n)}$.

Figure 4.7 shows an example of the domain decomposition algorithm in (4.15) for total variation inpainting. As for the 1D example in Figures 4.4-4.6 the operator T is a multiplier, i.e., $Tu = 1_{\Omega \setminus D} \cdot u$, where Ω denotes the rectangular image domain and $D \subset \Omega$ the missing domain in which the original image content got lost. The regularization parameter α is fixed at the value 10^{-2}. In Figure 4.7 the missing domain D is the black writing, which covers parts of the image. Here, the image domain of size 449×570 pixels is split into five overlapping subdomains with an overlap size $G = 28 \times 570$. Further, the fixed points η's are computed on a small stripe $\hat{\Omega}_i$, $i = 1, \ldots, 5$ respectively, of size 6×570 pixels.

Finally, in Figure 4.8 we illustrate the successful application of our domain decomposition algorithm in (4.15) for a compressed sensing problem. Here, we consider a medical-type image (the so-called *Logan-Shepp phantom*) and its reconstruction from only partial Fourier data. In this case the linear operator $T = S \circ \mathcal{F}$, where $\mathcal{F}$ denotes the $2D$ Fourier matrix and S is a *downsampling operator* which selects only a few frequencies as output. We minimize $\mathcal{J}$ with α set at 0.4×10^{-2}. In the application of the algorithm in (4.15) the image domain of size 256×256 pixels is split into four overlapping subdomains with an overlap size $G = 20 \times 256$. The fixed points η's are computed in a small stripe $\hat{\Omega}_i$, $i = 1, \ldots, 4$ respectively, of size 6×256


Initial Picture
146 iterations
dog on board dog on board dog
on board dog on board dog on
board dog on board dog on

(a) (b)

Figure 4.7: *This figure shows an application of the algorithm in* (4.15) *for image inpainting. In this simulation the problem was split into five subproblems on overlapping subdomains.*

pixels.

Remark 4.2.13. *The optimization of the different parameters of the algorithm, namely, the number of subdomains, the extent of the overlapping regions, the number of internal iterations, and the relationship with the rate of convergence are a very challenging problem, and a matter of current investigation.*

Numerical Experiments for the Parallel Implementation

In this section we show that already with relatively large images, classical methods do not scale well and may converge slowly, whereas the parallel implementation of the algorithm on a multiple processor computer allows for significant reductions of the CPU time, which improves with the number of subdomains. For a fair comparison we utilize Chambolle's algorithm both for the solution on the whole domain and on subdomains. We expect that the use of other algorithms [26, 46, 69, 96, 119] may change quantitatively the results but not qualitatively. Of course, considering problems of even larger size, for example, in higher dimension, can only further promote and favour the use of parallel strategies.

In Figure 4.9 we depict an image of size 5616×3744, which has been vandalized by superimposing a text. In Figure 4.10 we show the results due to the application of the parallel algorithm in (4.31) acting on 4 and 16 subdomains.

We do not dispose of a minimizer in this case (except for considering as an approximate minimizer one of the iterations $u^{(n)}$ for n very large) and the value of $\mathcal{J}$ is not a good indicator of the quality of image restoration. Being the scope of the minimization the approximate recovery of the original image, which we may denote u_{org}, we use as a stopping criterion $\|u^{(n)} - u_{org}\|_2^2 < \epsilon$, for a prescribed tolerance

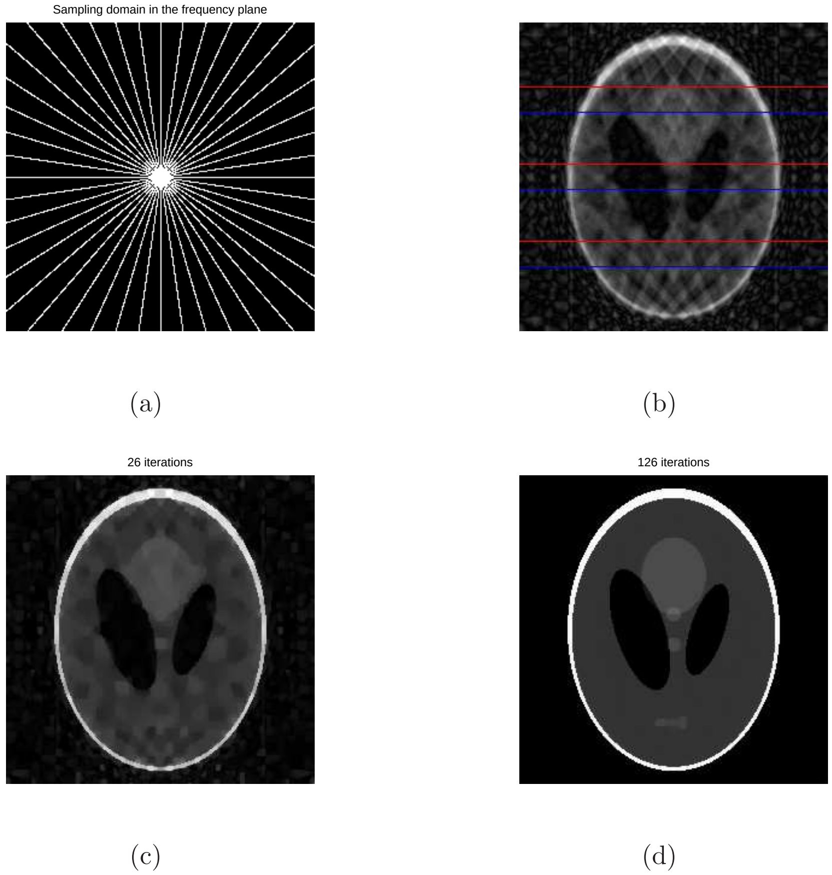


Figure 4.8: *We show an application of the algorithm in* (4.15) *in a classical compressed sensing problem for recovering piecewise constant medical-type images from given partial Fourier data. In this simulation the problem was split via decomposition into four overlapping subdomains. On the top-left figure, we show the sampling data of the image in the Fourier domain. On the top-right the back-projection provided by the sampled frequency data together with the highlighted partition of the physical domain into four subdomains is shown. The bottom figures present intermediate iterations of the algorithm, i.e.,* $u^{(26)}$ *and* $u^{(125)}$.

	1 domain	4 domains	16 domains
CPU time	23086.68 s	6531.94 s	1583.52 s
Nr. outer iterations	1000	10	10

Table 4.1: *Regularization parameter* $\alpha = 0.1$, 3 *inner iterations on the subdomains. The stopping criterion for all three algorithms is when the squared* L^2*-norm of the difference between the current minimizer and the original image* $\|u^{(n)} - u_{org}\|_2^2$ *gets below* $\epsilon = 0.0048$.

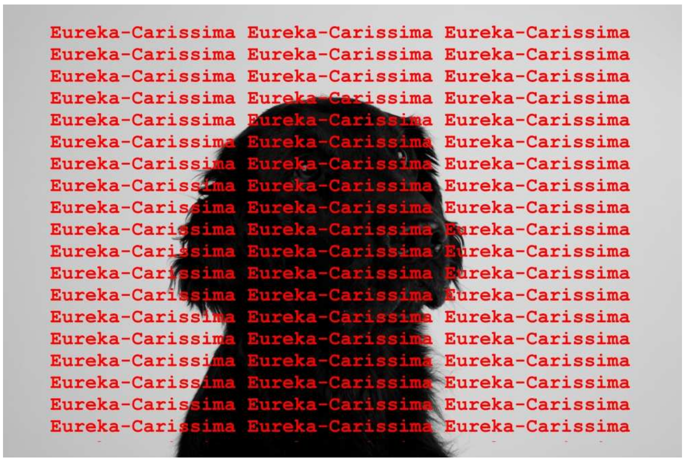

Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima
Eureka-Carissima Eureka-Carissima Eureka-Carissima

Figure 4.9: *Image size* 5616×3744*.*

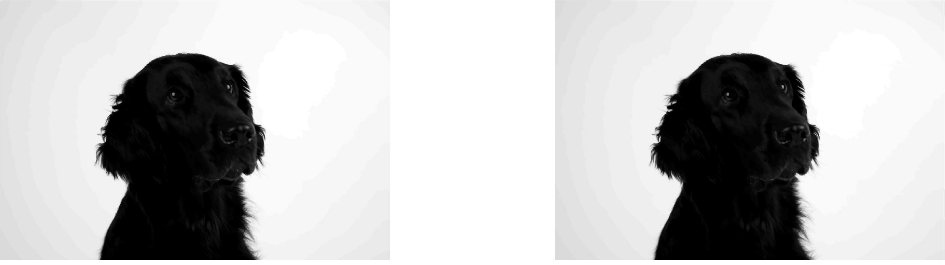


Figure 4.10: *The computed approximate minimizers for the regularization parameter* $\alpha = 0.1$*,* 100 *outer iterations and* 3 *inner iterations, for* 4 *domains (left picture) and* 16 *domains (right picture).*

ϵ. While the algorithm applied on the whole domain does not reach the prescribed accuracy after more than 6 hours of running time, the computation with multiple subdomains can reach the result in less than half an hour. We also emphasize that in these experiments the computational time decreases linearly with the number of subdomains, showing that the computation of the Lagrange multipliers, used in our algorithm in order to correctly interface the patches, has a nearly negligible cost with respect to the minimizations on the subdomains, see Table 4.1.

4.3 Non-overlapping Domain Decomposition Algorithm

The work presented in the previous section was particularly addressed to overlapping domain decomposition. In this section we show how to specify the subspace correction algorithm from Chapter 3, i.e., the algorithm in (3.13), to the case of a non-overlapping domain decomposition as suggested in [66]. The functional of interest to be minimized is again the discrete functional $\mathcal{J}$ in (4.1) together with the coercivity condition (**C**).

Now, instead of minimizing $\mathcal{J}$ on the whole domain, we propose to decompose Ω into disjoint and non-overlapping subdomains. We limit ourself to split the problem into two disjoint subdomains Ω_1 and Ω_2 such that $\Omega_1 \subset \Omega$ and $\Omega_2 = \Omega \setminus \Omega_1$, but one can easily generalize the splitting to multiple subdomains, see Remark 4.2.10. As in the previous section, we assume again, only for simplicity, that also the subdomains Ω_i are discrete d-orthotopes as well as Ω. Due to this domain decomposition $\mathcal{H}$ is split into two closed orthogonal and complementary subspaces $V_i = \{u \in \mathcal{H} : \operatorname{supp}(u) \subset \Omega_i\}$, for $i = 1, 2$, i.e., $\mathcal{H} = V_1 \oplus V_2$. Note that in the following $u_i = \pi_{V_i}(u)$, for $i = 1, 2$. Now we would like to solve (4.4) by picking an initial $V_1 \oplus V_2 \ni u_1^{(0)} + u_2^{(0)} := u^{(0)} \in \mathcal{H}$, e.g., $u_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} u_1^{(n+1)} \approx \arg\min_{v_1 \in V_1} \mathcal{J}(v_1 + u_2^{(n)}) \\ u_2^{(n+1)} \approx \arg\min_{v_2 \in V_2} \mathcal{J}(u_1^{(n+1)} + v_2) \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)}. \end{cases} \tag{4.35}$$

The subspace minimization problems of the algorithm in (4.35) are solved as described in Section 3.1. That is, for $\mathcal{J}_1^s$ defined as in (3.4) now with the spaces V_i from above, each subspace minimization is approximated by the surrogate functional minimization

$$u_1^{(0)} \in V_1, \quad u_1^{(\ell+1)} = \arg\min_{u_1 \in V_1} \mathcal{J}_1^s(u_1 + u_2, u_1^{(\ell)}), \quad \ell \geq 0$$

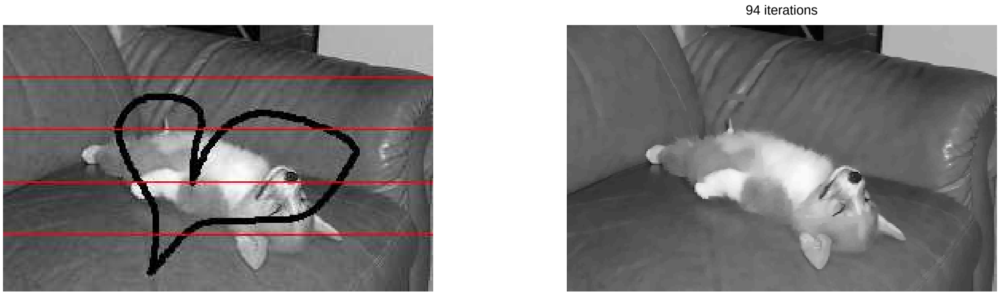
(cf. (3.5)), which is then solved by Lagrange multipliers or more precisely by iterative oblique thresholding, see Theorem 3.1.4.

4.3.1 Convergence of the Sequential Domain Decomposition Method

Let us return to the sequential algorithm in (3.13) and express it explicitly for the case of a non-overlapping domain decomposition as follows: pick an initial $V_1 \oplus V_2 \ni u_1^{(0,L)} + u_2^{(0,M)} := u^{(0)} \in \mathcal{H}$, e.g., $u_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = u_1^{(n,L)} \\ u_1^{(n+1,\ell+1)} = \arg\min_{u_1 \in V_1} \mathcal{J}_1^s(u_1 + u_2^{(n,M)}, u_1^{(n+1,\ell)}) \quad \ell = 0, \dots, L-1 \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = u_2^{(n,M)} \\ u_2^{(n+1,m+1)} = \arg\min_{u_2 \in V_2} \mathcal{J}_2^s(u_1^{(n+1,L)} + u_2, u_2^{(n+1,m)}) \quad m = 0, \dots, M-1 \end{cases} \\ u^{(n+1)} := u_1^{(n+1,L)} + u_2^{(n+1,M)}. \end{cases} \tag{4.36}$$

In this section we want to prove its convergence to a minimizer of the discrete functional $\mathcal{J}$ for any choice of finite numbers L and M of inner iterations. We recall that by Theorem 3.1.12 the algorithm in (4.36) decreases the energy $\mathcal{J}$ monotonically and converges. Moreover only under some technical conditions, which are in general not fulfilled, the algorithm even converges to a minimizer of the original functional (4.1). However, in the numerical experiments shown in [66], the algorithm seems always converging robustly to the expected minimizer, see Figure 4.11.



(a)

(b)

Figure 4.11: *This figure shows an application of the sequential domain decomposition algorithm in* (4.36) *for image inpainting. In this simulation the problem was split into five subproblems on non-overlapping subdomains. The interfaces of the subdomains are marked in red.*

Convergence to Minimizers

We close this gap between the lacking theoretical analysis and the promising numerical examples from above by showing that the algorithm in (4.36) indeed converges to an expected minimizer in our discrete setting. In order to do that, we use again Proposition 3.2.2 as a characterization of solutions of the minimization problem (4.4). Then by following the same strategy as in the proof of Theorem 4.2.8 we are eventually able to prove the convergence of the algorithm in (4.36) to minimizers of $\mathcal{J}$.

Theorem 4.3.1 (Convergence to minimizers)**.** *Assume $\|T\| < 1$. Then accumulation points of the sequence $(u^{(n)})_n$ produced by the algorithm in (4.36) are minimizers of $\mathcal{J}$. If $\mathcal{J}$ has a unique minimizer, then the sequence $(u^{(n)})_n$ converges to it.*

Proof. Note that due to the orthogonal splitting of Ω the sequences $(u_1^{(n,L)})_n$ and $(u_2^{(n,M)})_n$ produced by the algorithm in (4.36) are bounded. Hence there exist convergent subsequences, which we denote for ease again by $(u_1^{(n,L)})_n$ and $(u_2^{(n,M)})_n$. Let us denote by $u_1^{(\infty)}$ the limit of the sequence $(u_1^{(n,L)})_n$ and $u_2^{(\infty)}$ the limit of the sequence $(u_2^{(n,M)})_n$. Then by analogous arguments as the ones in the proof of Theorem 4.2.8 we obtain with the help of Proposition 3.2.2 the following optimality conditions

$$\begin{aligned} \langle \bar{M}_1^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| &= 0 \quad \text{for all } x \in \Omega \\ -2(u_1^{(\infty)}(x) - z_1^{(\infty)}(x)) - \operatorname{div} \bar{M}_1^{(\infty)}(x) - 2\eta_1^{(\infty)}(x) &= 0 \quad \text{for all } x \in \Omega \end{aligned}$$

for $u_1^{(\infty)}$ and the following optimality conditions

$$\begin{aligned} \langle \bar{M}_2^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| &= 0 \quad \text{for all } x \in \Omega \\ -2(u_2^{(\infty)}(x) - z_2^{(\infty)}(x)) - \operatorname{div} \bar{M}_2^{(\infty)}(x) - 2\eta_2^{(\infty)}(x) &= 0 \quad \text{for all } x \in \Omega \end{aligned}$$

for $u_2^{(\infty)}$.

Since $\eta_1^{(\infty)} \in V_2$ is only supported in Ω_2, i.e., $\eta_1^{(\infty)}(x) = 0$ in Ω_1, and $\eta_2^{(\infty)} \in V_1$ is only supported in Ω_1, i.e., $\eta_2^{(\infty)}(x) = 0$ in Ω_2, we have

$$\begin{aligned} \langle \bar{M}_1^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \quad \text{for all } x \in \Omega \\ -2\pi_{V_1} T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \bar{M}_1^{(\infty)}(x) &= 0 \quad \text{for all } x \in \Omega_1 \end{aligned} \tag{4.37}$$

$$\begin{aligned} \langle \bar{M}_2^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \quad \text{for all } x \in \Omega \\ -2\pi_{V_2} T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \bar{M}_2^{(\infty)}(x) &= 0 \quad \text{for all } x \in \Omega_2. \end{aligned} \tag{4.38}$$

Observe now that from Proposition 3.2.2 we also have that $0 \in \mathcal{J}(u^{(\infty)})$ if and only if there exists $M^{(\infty)} = (M_0^{(\infty)}, \bar{M}^{(\infty)})$ with $|\bar{M}^{(\infty)}(x)| \leq 2\alpha$ for all $x \in \Omega$ such that

$$\begin{aligned} \langle \bar{M}^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \quad \text{for all } x \in \Omega \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \bar{M}^{(\infty)}(x) &= 0 \quad \text{for all } x \in \Omega. \end{aligned} \tag{4.39}$$

Hence let us choose

$$M^{(\infty)}(x) = \begin{cases} M_1^{(\infty)}(x) & \text{if } x \in \Omega_1 \\ M_2^{(\infty)}(x) & \text{if } x \in \Omega_2 \end{cases}.$$

With this choice of $M^{(\infty)}$ equations (4.37) - (4.39) are valid and hence $u^{(\infty)}$ is optimal in Ω. □

Remark 4.3.2. *Note that in comparison to the proof of Theorem 4.2.8, here we could not use the overlapping property of the subdomains, but we took strongly advantage of the fact that* $\operatorname{supp} \eta_1 \subset \Omega_2$ *and* $\operatorname{supp} \eta_2 \subset \Omega_1$*. Hence we could restrict the corresponding optimality conditions in* (4.37) *and* (4.38) *to the domain* Ω_1 *and* Ω_2 *only.*

4.3.2 A Parallel Algorithm and its Convergence

The parallel version of the previous algorithm in (4.36) may be written as follows: pick an initial $V_1 \oplus V_2 \ni u_1^{(0,L)} + u_2^{(0,M)} := u^{(0)} \in \mathcal{H}$ and iterate

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = u_1^{(n,L)} \\ u_1^{(n+1,\ell+1)} = \arg\min_{u_1 \in V_1} \mathcal{J}_1^s(u_1 + u_2^{(n,M)}, u_1^{(n+1,\ell)}) \quad \ell = 0, \ldots, L-1 \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = u_2^{(n,M)} \\ u_2^{(n+1,m+1)} = \arg\min_{u_2 \in V_2} \mathcal{J}_2^s(u_1^{(n,L)} + u_2, u_2^{(n+1,m)}) \quad m = 0, \ldots, M-1 \end{cases} \\ u^{(n+1)} := \frac{u_1^{(n+1,L)} + u_2^{(n+1,M)} + u^{(n)}}{2}. \end{cases} \tag{4.40}$$

We get similar convergence properties of this algorithm as for (4.36).

Theorem 4.3.3. *The algorithm in (4.40) produces a sequence* $(u^{(n)})_n$ *in* $\mathcal{H}$ *with the following properties:*

(i) $\mathcal{J}(u^{(n)}) > \mathcal{J}(u^{(n+1)})$ *for all* $n \in \mathbb{N}$ *(unless* $u^{(n)} = u^{(n+1)}$*);*

(ii) $\lim_{n\to\infty} \|u^{(n+1)} - u^{(n)}\|_{\mathcal{H}} = 0$*;*

(iii) the sequence $(u^{(n)})_n$ *has subsequences which converge in* $\mathcal{H}$*.*

The proof of this theorem is analogue to the one of Proposition 4.2.11. Moreover we can show by similar arguments as in Theorem 4.2.12 the following statement.

Theorem 4.3.4 (Convergence to minimizers)**.** *Assume* $\|T\| < 1$*. Then accumulation points of the sequence* $(u^{(n)})_n$ *produced by algorithm (4.40) are minimizers of* $\mathcal{J}$*. If* $\mathcal{J}$ *has a unique minimizer, then the sequence* $(u^{(n)})_n$ *converges to it.*

4.4 Bregmanized Non-overlapping Domain Decomposition

In this section we are now concerned with increasing the performance of the non-overlapping domain decomposition algorithms introduced in Section 4.3, by using a more efficient technique to solve the subminimization problems. It turned out that we have to solve on each subdomain constrained optimization problems of the type

$$\min_{u\in\mathcal{H}}\{F(u) := \|u - z\|_2^2 + 2\alpha|\nabla u|(\Omega) \text{ subject to } Au = f\}, \tag{4.41}$$

where z and f are functions given on $\mathcal{H}$ and A is a linear bounded operator in $\mathcal{H}$. In particular, for the non-overlapping domain decomposition method the constraint is the orthogonal projection onto a subspace, while for the overlapping algorithm the linear constraint is simply a trace condition.

It is well-known that the *Augmented Lagrange Method* [72] and its variations known under the name of Bregman iterations [96, 123] solve (4.41) efficiently. Nevertheless in the previous sections we used the iterative oblique thresholding technique, see for example Section 3.1, to compute an approximate solution of (4.41). In this section, we suggest now to compute the solution of the subproblems by nested Bregman iterations. More precisely, we use the recently introduced *Bregmanized Operator Splitting* technique [124] combined with the *Split Bregman method* [69] for the solution of (4.41) and call this new method simply *Bregmanized Operator Splitting - Split Bregman* algorithm. This approach avoids the computation of a costly fixed point iteration in each domain decomposition step and consequently speeds up the overall computational time of the domain decomposition algorithms, cf. the numerical examples in Section 4.4.3.

4.4.1 Bregmanized Operator Splitting - Split Bregman Algorithm

The *Bregman distance*, associated with a convex functional $F : \mathcal{H} \to \bar{\mathbb{R}}$, of the vectors $u, v \in \mathrm{Dom}(F)$ is defined by

$$D_F^p(u, v) := F(u) - F(v) - \langle p, u - v\rangle,$$

for $p \in \partial F(v)$. Note that the Bregman distance is not a distance in the usual sense, since it is in general not symmetric and also the triangle inequality does not hold. However it satisfies $D_F^p(u, v) \geq 0$ and $D_F^p(u, v) = 0$ if $u = v$ [14].

In [96] the so-called Bregman Iteration was proposed in order to solve constrained optimization problems of the type (4.41):

Algorithm 1. *Bregman Iteration: Let $\lambda > 0$ and $u^{(0)} = 0$ then for $k = 0, 1, \ldots$ do*

$$\begin{aligned} p^{(k)} &\in \partial F(u^{(k)}) \\ u^{(k+1)} &= \arg\min_{u\in\mathcal{H}} D_F^{p^{(k)}}(u, u^{(k)}) + \lambda\|Au - f\|_2^2. \end{aligned} \tag{4.42}$$

The weak convergence of this algorithm to a solution of (4.41) is ensured and it is shown that the sequence of residuals $(\|Au^{(k)} - f\|)_k$ is monotonically decreasing to zero, see [96]. Since the Bregman Iteration is equivalent to the Augmented Lagrangian Method its convergence is also guaranteed by the results in [59]. Moreover, in [123] it has been shown that the Bregman Iteration is equivalent to the following simplified iterative scheme:

Algorithm 2. *Simplified Bregman Iteration: Let $\lambda > 0$. Initialize $u^{(0)} = 0$ and $f^{(0)} = 0$ then for $k = 0, 1, \dots$ do*

$$\begin{aligned} u^{(k+1)} &= \arg\min_{u\in\mathcal{H}} F(u) + \lambda\|Au - f - f^{(k)}\|_2^2 \\ f^{(k+1)} &= f^{(k)} - Au^{(k+1)}. \end{aligned} \tag{4.43}$$

The direct computation of the update $u^{(k+1)}$ in (4.42) and (4.43) is sometimes not efficiently and exactly solvable, in particular if the constraint is ill-posed. In order to overcome this drawback we may suggest to solve the minimization problem in (4.43) via a forward-backward operator splitting, see [39] for more details. In particular, we are interested in the Bregmanized Operator Splitting algorithm [124], which is based on one forward-backward operator splitting iteration and a suitable update of the Lagrange multiplier:

Algorithm 3. *Bregmanized Operator Splitting (BOS): Let $\lambda, \delta > 0$. Initialize $u^{(0)} = 0$ and $f^{(0)} = 0$ then for $k = 0, 1, \dots$ do*

$$\begin{aligned} u^{(k+1)} &= \arg\min_{u\in\mathcal{H}} F(u) + \frac{\lambda}{\delta}\|u - (u^{(k)} - \delta A^*(Au^{(k)} - f - f^{(k)}))\|_2^2 \\ f^{(k+1)} &= f^{(k)} - Au^{(k+1)}. \end{aligned} \tag{4.44}$$

This algorithm is ensured to converge to a minimal solution of (4.41) if $0 < \delta < \frac{1}{\|A^*A\|}$. Moreover, it is very stable in practice, and it is usually easy to implement.

We note that the minimization problem in (4.44) is equivalent to the famous ROF-problem [103] and therefore there exist several numerical methods that solve this problem efficiently, see [23, 26, 46, 69]. Among the fastest is the Split Bregman Method [69], whose main idea is to consider instead of

$$\arg\min_{u\in\mathcal{H}} \|u - z\|_2^2 + 2\alpha|\nabla u|(\Omega) + \frac{\lambda}{\delta}\|u - (u^{(k)} - \delta A^*(Au^{(k)} - f - f^{(k)}))\|_2^2$$

the following equivalent constrained problem

$$\arg\min_{u,d} \|u - z\|_2^2 + 2\alpha|d|(\Omega) + \frac{\lambda}{\delta}\|u - (u^{(k)} - \delta A^*(Au^{(k)} - f - f^{(k)}))\|_2^2 \quad \text{s.t.} \quad d = \nabla u.$$

Solving this constrained minimization problem by the simplified Bregman Iteration we get the Split Bregman Method:

$$\begin{aligned}(u^{(l+1)},d^{(l+1)}) &= \arg\min_{u,d}\|u-z\|_2^2+2\alpha|d|(\Omega)+\frac{\lambda}{\delta}\|u-(u^{(k)}-\delta A^*(Au^{(k)}-f-f^{(k)}))\|_2^2\\ &\qquad\qquad\qquad\qquad\qquad\qquad+\mu\|d-\nabla u-b^{(l)}\|_2^2\\ b^{(l+1)}&=b^{(l)}+(\nabla u^{(l+1)}-d^{(l+1)}),\end{aligned}\tag{4.45}$$

where $\mu > 0$. We propose to combine the Bregmanized Operator Splitting with the Split Bregman Iteration to solve (4.41), which results in an algorithm using two nested iterations:

Algorithm 4. *Bregmanized Operator Splitting - Split Bregman (BOS-SB): Let* $\lambda, \delta, \mu > 0$ *be regularization parameters. Initialize* $u^{(0,L_0)} = 0$ *and* $f^{(0)} = 0$ *then for* $k = 0, 1, \ldots$ *do*

$$\left\{\begin{array}{l} u^{(k+1,0)} = u^{(k,L_k)}, d^{(k+1,0)} = b^{(k+1,0)} = 0\\ \textit{for } l = 0,\ldots,L_{k+1} \textit{ do}\\ \left\{\begin{array}{l} u^{(k+1,l+1)} = \arg\min_{u\in\mathcal{H}}\|u-z\|_2^2\\ \qquad\qquad +\frac{\lambda}{\delta}\|u-(u^{(k,L_k)}-\delta A^*(Au^{(k,L_k)}-f-f^{(k)}))\|_2^2\\ \qquad\qquad +\mu\|d^{(k+1,l)}-\nabla u-b^{(k+1,l)}\|_2^2\\ d^{(k+1,l+1)} = \arg\min_d 2\alpha|d|(\Omega)+\mu\|d-\nabla u^{(k+1,l+1)}-b^{(k+1,l)}\|_2^2\\ b^{(k+1,l+1)} = b^{(k+1,l)}+\nabla u^{(k+1,l+1)}-d^{(k+1,l+1)} \end{array}\right.\\ f^{(k+1)} = f^{(k)} - Au^{(k+1,L_{k+1})}. \end{array}\right.\tag{4.46}$$

The number of inner iteration L_k is chosen such that $\|u^{(k,L_k)} - u^{(k,L_k-1)}\|_2 \leq tol$.

4.4.2 Solution of the Subspace Minimization Problems

In this section we discuss the minimization of functional (4.1) by using the non-overlapping domain decomposition approach suggested in Section 4.3 and we propose to solve the corresponding subminimization problems with the help of Algorithm 4. In order to address the subminimization problems of the algorithm in (4.36) we have to solve

$$u_i^{(\ell+1)} = \arg\min_{u_i\in V_i}\mathcal{J}_i^s(u_1+u_2,u_i^{(\ell)}) = \arg\min_{u_i\in V_i}\|u_i-z_i\|_2^2+2\alpha|\nabla(u_1+u_2)|(\Omega)\quad \ell\geq 0,\tag{4.47}$$

where $z_1 = u_1^{(\ell)} + \pi_{V_1}T^*(g - Tu_2 - Tu_1^{(\ell)})$ and $z_2 = u_2^{(\ell)} + \pi_{V_2}T^*(g - Tu_1 - Tu_2^{(\ell)})$.

Let us further decompose $\Omega_2 = \hat{\Omega}_2 \cup (\Omega_2 \setminus \hat{\Omega}_2)$ with $\partial\hat{\Omega}_2 \cap \partial\Omega_1 = \partial\Omega_2 \cap \partial\Omega_1$, where $\hat{\Omega}_2 \subset \Omega_2$ is a neighborhood stripe around the interface $\partial\Omega_2\cap\partial\Omega_1$. Analogously we split $\Omega_1 = \hat{\Omega}_1 \cup (\Omega_1 \setminus \hat{\Omega}_1)$ with $\partial\hat{\Omega}_1 \cap \partial\Omega_2 = \partial\Omega_1 \cap \partial\Omega_2$. Associated to these decompositions we define $\hat{V}_i = \{u \in \mathcal{H} : \mathrm{supp}(u) \subset \hat{\Omega}_i\}$. By the splitting of the total variation (2.6), i.e.,

$$|\nabla(u_1+u_2)|(\Omega) = |\nabla(u_{1|_{\Omega_1\cup\hat{\Omega}_2}} + u_{2|_{\Omega_1\cup\hat{\Omega}_2}})|(\Omega_1\cup\hat{\Omega}_2) + |\nabla u_{2|_{\Omega_2\setminus\hat{\Omega}_2}}|(\Omega_2\setminus\hat{\Omega}_2) + \int_{\partial\hat{\Omega}_2\cap\partial(\Omega_2\setminus\hat{\Omega}_2)} |u_2^+ - u_2^-| d\mathcal{H}_{d-1}(x), \tag{4.48}$$

where $\mathcal{H}_d$ is the Hausdorff measure of dimension d and $u_{|_{\Omega_1\cup\hat{\Omega}_2}}$ is the restriction of u to $\Omega_1\cup\hat{\Omega}_2$, we can restrict the minimization in (4.47) to the domain $\Omega_1\cup\hat{\Omega}_2$ and $\Omega_2\cup\hat{\Omega}_1$ respectively, i.e.,

$$\begin{aligned} u_1^{(\ell+1)} &= \arg\min_{u_1\in V_1} \|u_1 - (u_1^{(\ell)} + \pi_{V_1}T^*(g - Tu_2 - Tu_1^{(\ell)}))\|_2^2 \\ &\quad + 2\alpha \left|\nabla(u_{1|_{\Omega_1\cup\hat{\Omega}_2}} + u_{2|_{\Omega_1\cup\hat{\Omega}_2}})\right|(\Omega_1\cup\hat{\Omega}_2), \quad \ell\geq 0, \\ u_2^{(\ell+1)} &= \arg\min_{u_2\in V_2} \|u_2 - (u_2^{(\ell)} + \pi_{V_2}T^*(g - Tu_1 - Tu_2^{(\ell)}))\|_2^2 \\ &\quad + 2\alpha \left|\nabla(u_{1|_{\Omega_2\cup\hat{\Omega}_1}} + u_{2|_{\Omega_2\cup\hat{\Omega}_1}})\right|(\Omega_2\cup\hat{\Omega}_1), \quad \ell\geq 0. \end{aligned}$$

Eventually, these subminimization problems can be rewritten as problems on $V_i\oplus\hat{V}_{\hat{i}}$, $i\in\{1,2\}$ and read

$$\arg\min_{u\in V_1\oplus\hat{V}_2} \|u - z_1\|_2^2 + 2\alpha\left|\nabla(u_{|_{\Omega_1\cup\hat{\Omega}_2}} + u_{2|_{\Omega_1\cup\hat{\Omega}_2}})\right|(\Omega_1\cup\hat{\Omega}_2) \quad \text{s.t.} \quad \pi_{\hat{V}_2}u = 0, \tag{4.49}$$

$$\arg\min_{u\in V_2\oplus\hat{V}_1} \|u - z_2\|_2^2 + 2\alpha\left|\nabla(u_{1|_{\Omega_2\cup\hat{\Omega}_1}} + u_{|_{\Omega_2\cup\hat{\Omega}_1}})\right|(\Omega_2\cup\hat{\Omega}_1) \quad \text{s.t.} \quad \pi_{\hat{V}_1}u = 0, \tag{4.50}$$

where $z_1 = u_1^{(\ell)} + \pi_{V_1}T^*(g - Tu_2 - Tu_1^{(\ell)})$ and $z_2 = u_2^{(\ell)} + \pi_{V_2}T^*(g - Tu_1 - Tu_2^{(\ell)})$. We write the constrained minimization problems (4.49) and (4.50) in another way. More precisely, we consider the problems

$$\arg\min_{\xi_1\in V_1\oplus\hat{V}_2} \|\xi_1 - u_2 - z_1\|_2^2 + 2\alpha\left|D(\xi_{1|_{\Omega_1\cup\hat{\Omega}_2}})\right|(\Omega_1\cup\hat{\Omega}_2) \quad \text{s.t.} \quad \pi_{\hat{V}_2}\xi_1 = u_2 \tag{4.51}$$

$$\arg\min_{\xi_2\in V_2\oplus\hat{V}_1} \|\xi_2 - u_1 - z_2\|_2^2 + 2\alpha\left|D(\xi_{2|_{\Omega_2\cup\hat{\Omega}_1}})\right|(\Omega_2\cup\hat{\Omega}_1) \quad \text{s.t.} \quad \pi_{\hat{V}_1}\xi_2 = u_1 \tag{4.52}$$

and note that for $i = 1,2$ and $\hat{i} = \{1,2\}\setminus\{i\}$ indeed ξ_i is optimal if and only if $u_i = \xi_i - u_{\hat{i}}$ is optimal. Moreover, notice that the new problems (4.51) and (4.52) are now of the type (4.41).

Bregmanized Operator Splitting - Split Bregman (BOS-SB)

In Section 4.3 the constrained minimization problems (4.49)-(4.50) are solved by oblique thresholding, see Section 3.2.6 for more details. The oblique thresholding iteration can be very slow in general, cf. [80]. Let us explain how we can accelerate this computation by replacing the oblique thresholding by the BOS-SB algorithm.

In order to speed up the computation of the algorithm in (4.35) we suggest to solve each of its subproblems by using Algorithm 4. Actually by Algorithm 4 we can directly compute a solution of the constrained optimization problems (4.51) and (4.52). That is, for example, the minimizer for (4.51) is computed by the following algorithm: let $A = \pi_{\hat{V}_2}$ and $\lambda, \delta, \mu > 0$ be regularization parameters. Initialize $\xi_1^{(0,L_0)} = \xi_1^{(\ell)} = u_1^{(\ell)} + u_2^{(\ell)}$ and $f^{(0)} = 0$ then for $k = 0, 1, \ldots$ do

$$\left\{ \begin{array}{l} \xi_1^{(k+1,0)} = \xi_1^{(k,L_k)}, d^{(k+1,0)} = b^{(k+1,0)} = 0 \\ \text{for } l = 0, \ldots, L_{k+1} \text{ do} \\ \left\{ \begin{array}{ll} \xi_1^{(k+1,l+1)} & = \arg\min_{\xi_1 \in V_1 \oplus \hat{V}_2} \frac{1}{2\alpha} \|\xi_1 - u_2 - z_1\|_2^2 \\ & \quad + \frac{\lambda}{\delta} \|\xi_1 - (\xi_1^{(k,L_k)} - \delta A^*(A\xi_1^{(k,L_k)} - u_2 - f^{(k)}))\|_2^2 \\ & \quad + \mu \|d^{(k+1,l)} - D(\xi_{1|_{\Omega_1 \cup \hat{\Omega}_2}}) - b^{(k+1,l)}\|_2^2 \\ d^{(k+1,l+1)} & = \arg\min_d |d|(\Omega_1 \cup \hat{\Omega}_2) + \mu \|d - D(\xi^{(k+1,l+1)}_{1|_{\Omega_1 \cup \hat{\Omega}_2}}) - b^{(k+1,l)}\|_2^2 \\ b^{(k+1,l+1)} & = b^{(k+1,l)} + D(\xi^{(k+1,l+1)}_{1|_{\Omega_1 \cup \hat{\Omega}_2}}) - d^{(k+1,l+1)} \end{array} \right. \\ f^{(k+1)} = f^{(k)} - A\xi_1^{(k+1,L_{k+1})}. \end{array} \right. \tag{4.53}$$

Then by setting $u_1^{(k+1,L_{k+1})} = \xi_1^{(k+1,L_{k+1})} - u_2$ for all $k = 0, 1, \ldots$ we obtain a sequence $(u_1^{(k,L_k)})_k$, which is converging to a solution of (4.49). Instead of making a detour by computing the sequence $(\xi_1^{(k,L_k)})_k$ we would like to find a solution of (4.49), i.e., the update $u_1^{(\ell+1)}$, directly. Therefore we note that $\xi_1^{(k+1,l+1)}$ is a minimizer of

$$\arg \min_{\xi_1 \in V_1 \oplus \hat{V}_2} \frac{1}{2\alpha} \|\xi_1 - u_2 - z_1\|_2^2 + \frac{\lambda}{\delta} \|\xi_1 - (\xi_1^{(k,L_k)} - \delta A^*(A\xi_1^{(k,L_k)} - u_2 - f^{(k)}))\|_2^2 \\ + \mu \|d^{(k+1,l)} - D(\xi_{1|_{\Omega_1 \cup \hat{\Omega}_2}}) - b^{(k+1,l)}\|_2^2$$

if and only if $u_1^{(k+1,l+1)} = \xi_1^{(k+1,l+1)} - u_2$ is a minimizer of

$$\arg \min_{u_1 \in V_1 \oplus \hat{V}_2} \frac{1}{2\alpha} \|u_1 - z_1\|_2^2 + \frac{\lambda}{\delta} \|u_1 - (u_1^{(k,L_k)} - \delta A^*(Au_1^{(k,L_k)} - f^{(k)}))\|_2^2 \\ + \mu \|d^{(k+1,l)} - D(u_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_{2|_{\Omega_1 \cup \hat{\Omega}_2}}) - b^{(k+1,l)}\|_2^2,$$

where $u_1^{(k,L_k)} = \xi_1^{(k,L_k)} - u_2$. By this observation and by the fact that $f^{(k)} = f^{(0)} + \sum_{i=1}^k u_2 - A\xi_1^{(i,L_i)} = f^{(0)} - \sum_{i=1}^k Au_1^{(i,L_i)}$ we can directly compute the update $u_1^{(\ell+1)}$ by the following algorithm: let $A = \pi_{\hat{V}_2}$ and $\lambda, \delta, \mu > 0$ be regularization parameters.

Initialize $u_1^{(0,L_0)} = u_1^{(\ell)}$ and $f^{(0)} = 0$ then for $k = 0, 1, \dots$ do

$$\left\{\begin{array}{l}
u_1^{(k+1,0)} = u_1^{(k,L_k)}, d^{(k+1,0)} = b^{(k+1,0)} = 0 \\
\text{for } l = 0, \dots, L_{k+1} \text{ do} \\
\left\{\begin{array}{l}
u_1^{(k+1,l+1)} = \arg\min_{u_1 \in V_1 \oplus \hat{V}_2} \frac{1}{2\alpha} \|u_1 - z_1\|_2^2 \\
\qquad\qquad + \frac{\lambda}{\delta} \|u_1 - (u_1^{(k,L_k)} - \delta A^*(A u_1^{(k,L_k)} - f^{(k)}))\|_2^2 \\
\qquad\qquad + \mu \|d^{(k+1,l)} - \nabla(u_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_{2|_{\Omega_1 \cup \hat{\Omega}_2}}) - b^{(k+1,l)}\|_2^2 \\
d^{(k+1,l+1)} = \arg\min_d |d|(\Omega_1 \cup \hat{\Omega}_2) + \mu \|d - \nabla(u^{(k+1,l+1)}_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_{2|_{\Omega_1 \cup \hat{\Omega}_2}}) - b^{(k+1,l)}\|_2^2 \\
b^{(k+1,l+1)} = b^{(k+1,l)} + \nabla(u^{(k+1,l+1)}_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_{2|_{\Omega_1 \cup \hat{\Omega}_2}}) - d^{(k+1,l+1)}
\end{array}\right. \\
f^{(k+1)} = f^{(k)} - A u_1^{(k+1,L_{k+1})}.
\end{array}\right. \tag{4.54}$$

In our finite dimensional setting the BOS-SB iteration (4.54) enjoys the following convergence properties.

Proposition 4.4.1. *Let* $u_1^{(k+1,0)} = u_1^{(k,L_k)} \in V_1$, $d^{(k+1,0)} = 0$ *and* $(u_1^{(k+1,l)})_l$ *and* $(d^{(k+1,l)})_l$ *be sequences generated by the Split Bregman Iteration* (4.45)*. Then* $u_1^{(k+1,l)} \to u_1^{(k+1,*)}$ *and* $d^{(k+1,l)} \to d^{(k+1,*)} = \nabla u_1^{(k+1,*)}$ *for* $l \to \infty$*, where* $u_1^{(k+1,*)}$ *solves the minimization problem given by one iteration of the Bregmanized Operator Splitting, i.e.,*

$$\begin{aligned}
\arg \min_{u_1 \in V_1 \oplus \hat{V}_2} \frac{1}{2\alpha} \|u_1 - z_1\|_2^2 + \frac{\lambda}{\delta} \|u_1 - (u_1^{(k,L_k)} - \delta A^*(A u_1^{(k,L_k)} - f^{(k)}))\|_2^2 \\
+ \left| \nabla(u_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_{2|_{\Omega_1 \cup \hat{\Omega}_2}}) \right| (\Omega_1 \cup \hat{\Omega}_2).
\end{aligned} \tag{4.55}$$

The proof is analogue to the one in [107], where the convergence of the Split Bregman Iteration is shown.

If we solve the minimization problem in (4.55) exactly, for example, via the Split Bregman Algorithm by setting formally $L_{k+1} = \infty$, then the following convergence property for Algorithm 4 holds:

Proposition 4.4.2. *Let* $u_1^{(k,*)}$ *be the exact solution of the minimization problem in* (4.55) *for the k-th iteration. Then the sequence* $(u_1^{(k,*)})_k$ *generated by* (4.54) *converges for* $k \to \infty$ *to a solution* u_1^* *of* (4.49)*.*

The proof of this result can be found in [124].

4.4.3 Numerical Examples for Image Restoration

We present the implementation of the algorithm in (4.36) and of the algorithm in (4.40) for the minimization of $\mathcal{J}$. To solve its subiterations (4.47) we consider two approaches: oblique thresholding, see Section 3.2.6, and the proposed BOS-SB iteration. In the following we sketch the numerical implementation of the BOS-SB algorithm for the domain Ω_1 only, since the implementation is analogue for the other domain by just adjusting the notations accordingly. Hence we denote $u_2 = u^{(n,M)}_{2|_{\Omega_1 \cup \hat{\Omega}_2}}$, $u_1 = u_1^{(n+1,\ell+1)}$, and $z_1 = u_1^{(n+1,\ell)} + \pi_{V_1} T^*(g - Tu_2 - Tu_1^{(n+1,\ell)})$ and we would like to compute the minimizer

$$u_1 = \arg\min_{u \in V_1} \|u - z_1\|_2^2 + 2\alpha |\nabla(u + u_2)|(\Omega_1 \cup \hat{\Omega}_2). \tag{4.56}$$

BOS-SB

Instead of using the oblique thresholding strategy we suggest to use Algorithm 4 to solve the minimization problem (4.56). Hence the minimizer u_1 is the limit of the sequence $(u_1^{(k,L_k)})_k$ generated by the algorithm in (4.54). The inner iteration of this algorithm is the Split Bregman iteration, i.e.,

$$\begin{cases} u_1^{(k+1,l+1)} = \arg \min\limits_{u_1 \in V_1 \oplus \hat{V}_2} \frac{1}{2\alpha}\|u_1 - z_1\|_2^2 + \frac{\lambda}{\delta}\|u_1 - (u_1^{(k,L_k)} - \delta A^*(Au_1^{(k,L_k)} - f^{(k)}))\|_2^2 \\ \qquad\qquad\quad +\mu\|d^{(k+1,\ell)} - \nabla(u_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_2) - b^{(k+1,l)}\|_2^2 \\ d^{(k+1,l+1)} = \arg\min_d |d|_{\ell_1(\Omega_1 \cup \hat{\Omega}_2)} + \mu\|d - \nabla(u^{(k+1,l+1)}_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_2) - b^{(k+1,l)}\|_2^2 \\ b^{(k+1,l+1)} = b^{(k+1,l)} + \nabla(u^{(k+1,l+1)}_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_2) - d^{(k+1,l+1)} \end{cases}$$

and is implemented as suggested in [69], with a small adaptation, since the distributional derivative of the sum of functions, i.e., $\nabla(u_{1|_{\Omega_1 \cup \hat{\Omega}_2}} + u_2)$, has to be considered on each subdomain.

Domain decompositions

Sequential algorithm For the sequential algorithm we split the domain $\Omega \subset \mathbb{R}^2$, i.e., the set of $N_1 \times N_2$ equidistant points on the 2-dimensional rectangle $[a,b] \times [c,d]$, into horizontal stripes, i.e., with respect to its rows. In particular we have $\Omega_1 = [a, x_{\lceil \frac{N_1}{2} \rceil}] \times [c,d]$ and $\Omega_2 = [x_{\lceil \frac{N_1}{2} \rceil + 1}, b] \times [c,d]$, compare Figure 4.12. The splitting in more than two domains is done similarly, cf. also [66].

In both subminimization strategies we additionally have to introduce the stripes $\hat{\Omega}_1$ and $\hat{\Omega}_2$ on the interfaces of the domain patches. These stripes arise naturally from the splitting of the total variation (4.48). In case of oblique thresholding, the stripe $\hat{\Omega}_1 \cup \hat{\Omega}_2$ defines the domain in which the η-computation (3.8) takes place. This is motivated by the observation that η is only supported on Ω_2 and that the, due to the restriction to this strip, produced errors are in practice negligible, cf. [66] for more details. In the other case when we use Algorithm 4 we just expand Ω_1 by the domain $\hat{\Omega}_1$, in which an additional constraint is constituted, see (4.49) and (4.50).

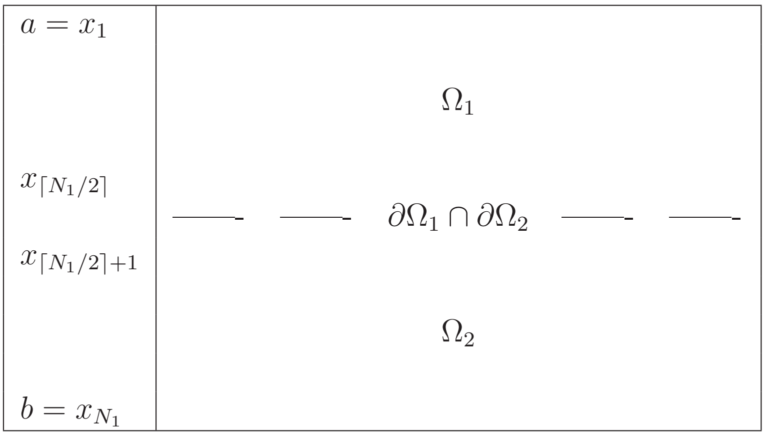

$a = x_1$
Ω_1
$x_{\lceil N_1/2 \rceil}$
$\partial\Omega_1 \cap \partial\Omega_2$
$x_{\lceil N_1/2 \rceil + 1}$
Ω_2
$b = x_{N_1}$

Figure 4.12: *Decomposition of the discrete image in two domains Ω_1 and Ω_2 with interface $\partial\Omega_1 \cap \partial\Omega_2$*

Parallel algorithm The splitting in the parallel version of the algorithm is done differently. While the choice of a different splitting for the parallel version has no theoretical motivation (in fact all the theory holds true for arbitrary domain splittings), it shows that the domain decomposition algorithm and its implementation are quite flexible with respect to the type of domain splitting.

For the parallel algorithm the discrete domain $\Omega = [a, b] \times [c, d]$ is split with respect to its rows and its columns. More precisely, we split Ω into powers of 4 rectangles, i.e., into $4, 16, 64, \ldots$ rectangles. This is done as shown in Figure 4.13. Note, that it is necessary to expand each of the subdomains by stripes around the interfaces (as in the previous section).

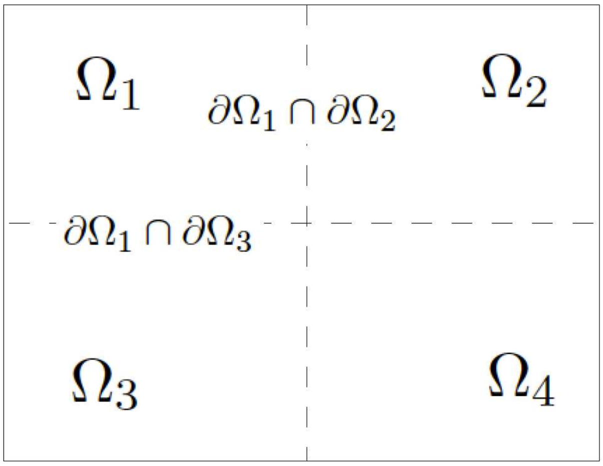

Ω_1
$\partial\Omega_1 \cap \partial\Omega_2$
Ω_2
$\partial\Omega_1 \cap \partial\Omega_3$
Ω_3
Ω_4

Figure 4.13: *Domain decomposition for the parallel algorithm in four subdomains Ω_i, $i = 1, 2, 3, 4$. The stripes for the subminimization on Ω_1 are located around the interfaces $\partial\Omega_1 \cap \partial\Omega_2$ and $\partial\Omega_1 \cap \partial\Omega_3$.*

Numerical Evaluation

We conclude this section with a numerical evaluation of the performance of the newly proposed subminimization strategy in Algorithm 4, i.e, *Bregmanized Operator Splitting - Split Bregman* (BOS-SB). To do so, we compare both the sequential- and the parallel version of the non-overlapping domain decomposition algorithm, where the subminimization problems are solved with iterative oblique thresholding (OT), see Section 3.2.6, and with BOS-SB (4.54) as a subminimization solver respectively. Here, we focus on its application to image inpainting. In this case the operator T is given by the multiplier $T = 1_{\Omega \setminus D}$, where D is a hole in the given image.

Let us first discuss the choice of the different parameters. In image inpainting the regularizing parameter α is typically chosen very small, in order to reduce the smoothing effect outside of the hole D. Hence, for the following inpainting examples we have chosen $\alpha = 0.005$. In the domain decomposition algorithm, we consider domain splittings into $\mathcal{N} = 2, 3, 4, 5$ for the sequential version, and $\mathcal{N} = 4, 16$ for the parallel version. The domain decomposition algorithm in (4.35) is iterated until the error $e^{(n)} = \|u_{org} - u^{(n)}\|_2 / \|u^{(n)}\|_2$ is smaller than a certain tolerance tol, where u_{org} is the original image. While this stopping criterion is rather unrealistic for practical applications, it serves us as a good basis for comparing the computational behaviour of the subminimization solvers. Note however, that in our experience the choice of this tolerance has a slightly different impact on the domain decomposition algorithm when we solve the subminimization problems with OT rather than BOS-SB. We shall discuss this in more detail when presenting the numerical examples.

The number of subminimization iterations L and M has been chosen to optimize the computation time for both OT and BOS-SB. In each subdomain we choose the same number of subiteration, i.e., $L = M =: sub$. For OT it turned out that $sub = 1$ is optimal (see also [66]). For BOS-SB $sub = 1$ or $sub = 4$ subminimization iterations give similar computational results in the sequential algorithm (see Table 4.4), while $sub = 4$ performs noticeably better in the parallel version of the algorithm. The reasons seem to be that each BOS-SB computation is much cheaper in terms of computational time than the OT solution (cf. Table 4.2), but also that the BOS-SB computation makes more progress in terms of decreasing the error $e^{(n)}$ in each subspace iteration than OT does (cf. Figure 4.19).

Next, we discuss the parameter choice taken for the subminimization algorithms. We start with oblique thresholding.

Parameter choice for oblique thresholding (OT) The choice of parameters in the oblique thresholding algorithm and their reasoning is discussed in much detail in [66]. In particular, in [66] this choice has been optimized with respect to the computational efficiency of the domain decomposition algorithm, for both its sequential (4.36) and its parallel (4.40) version. Therefore, we borrow the parameter values from there and only report them here. The width of the stripe in which the Lagrange multiplier η is computed is taken equal to 6 (this can be decreased or increased depending on the size of the regularization parameter α; again see [66] for a discussion on this). The fixed point algorithm for η either terminates when the normalized ℓ_2-distance between two subsequent iterates is smaller than $tol_\eta = 10^{-6}$ or after a maximal number of 10 iterations. The fixed point algorithm of Chambolle [23] for the computation of the projection $P_{\alpha K}(\cdot)$ terminates when the normalized ℓ_2-distance between two subsequent iterates is smaller than $tol_p = 10^{-3}$.

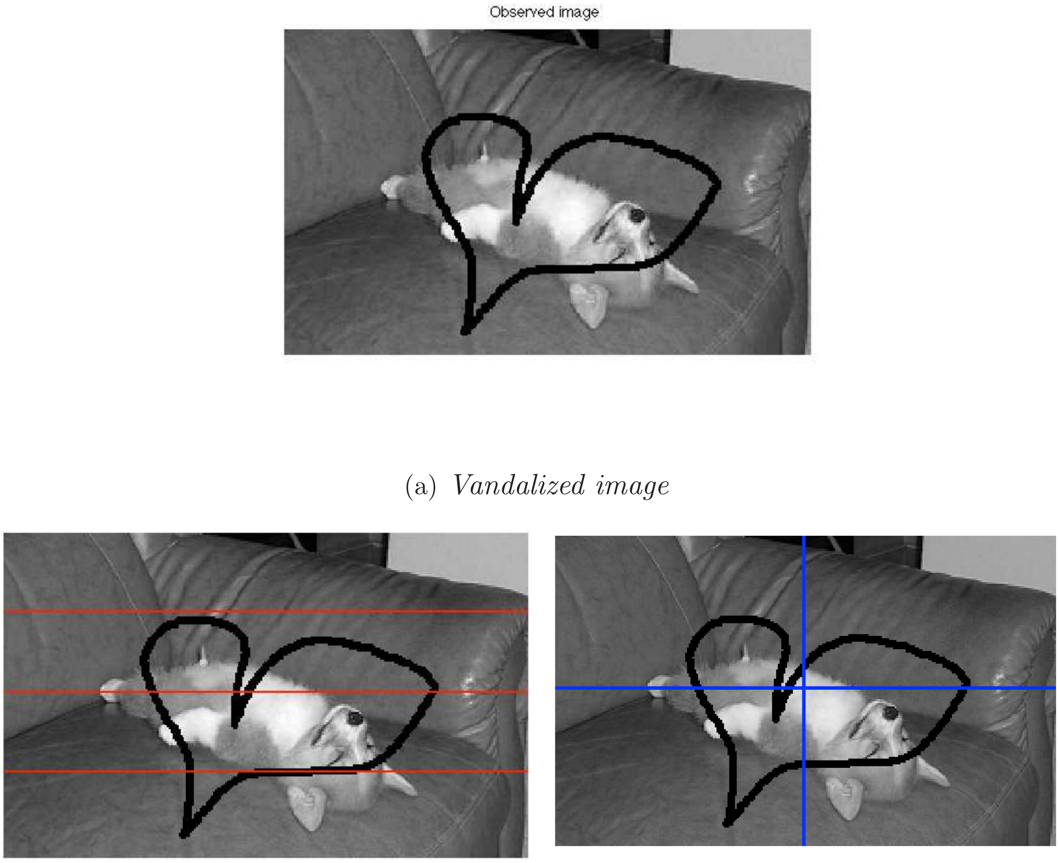
Parameter choice for Bregmanized operator splitting - split Bregman (BOS-SB) Algorithm 4, i.e., BOS-SB, consists of two nested iterations. The outer iteration is the Bregmanized operator splitting (BOS) iteration Algorithm 3, in which the corresponding minimization problem in each iteration is solved via the Split Bregman (SB) algorithm (4.45) that is again solved iteratively (inner iteration).

The number of BOS-iterations has been chosen equal to 1. In fact, our numerical tests confirm that there is absolutely no gain in terms of computational performance when iterating more. In particular, the number of domain decomposition iterations undertaken to reach a certain accuracy $e^{(n)} < tol$ is exactly the same when iterating BOS once or iterating twice or more. The reason for this is that we chose the parameter λ/δ in front of the Bregman fidelity term $\|u - (u^{(k,L_k)} - \delta A^*(Au^{(k,L_k)} - f^{(k)}))\|_2^2$ very small, i.e., $\lambda/\delta = 10^{-8}/2$. Although this means that we ensure the constraint $Au = 0$ very loosely only, this is adjusted by the reconsideration of this constraint in every domain decomposition iteration. Moreover, note that the BOS-iterations are proven to converge for every choice of the parameters $\lambda, \delta > 0$, cf. Proposition 4.4.2.

The Split Bregman algorithm (4.45) is solved with $\mu = 10$ and iterated until the normalized ℓ_2-distance of two subsequent iterates $u^{(l)}$ and $u^{(l+1)}$ is smaller than 10^{-3}. This choice has been made comparable to the tolerance for the Chambolle algorithm in the previous paragraph.

Numerical results - sequential algorithm We present numerical results for the sequential and the parallel version of the domain decomposition algorithm in (4.35), in which we compare the performance of OT and BOS-SB in terms of quality of the inpainting results and the computational time needed to achieve them. The numerical examples presented here have been computed on a 2×3.2 GHz-Quad-Core MacPro. In Figure 4.14(a) we start our numerical discussion with an inpainting task for an image of size 270×167. The decompositions of the image domain into $\mathcal{N}$ subdomains are done differently for the sequential and the parallel version of the algorithm, see above. For further reference, we plotted the decompositions for the image in Figure 4.14(a) in $\mathcal{N} = 4$ domains for the sequential and parallel case in Figure 4.14(b) and 4.14(c) respectively.

In Figure 4.15 we apply the sequential domain decomposition algorithm in (4.36) with $\mathcal{N} = 5$ subdomains for inpainting the image in 4.14(a). The inpainting results computed with OT and with BOS-SB are presented in Figure 4.15(a) and 4.15(b) respectively.

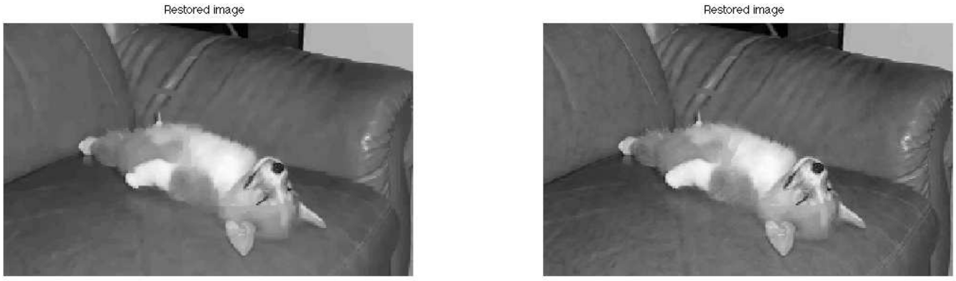


(a) *Vandalized image*

(b) *Splitting for the sequential algorithm*

(c) *Splitting for the parallel algorithm*

Figure 4.14: *The vandalized image and its domain splitting for the sequential and the parallel version of the algorithm for $\mathcal{N} = 4$ domains.*



(a) *Image inpainted with OT* (b) *Image inpainted with BOS-SB*

Figure 4.15: *Inpainting with the* sequential version *of domain decomposition* (4.36) *in* $\mathcal{N} = 5$ *subdomains: (a) inpainted image with OT used to solve the subminimization problems; (b) inpainted image with BOS-SB used to solve the subminimization problems.*

For a first comparison between the two algorithms, we report their computational speed for solving one subspace minimization averaged over the first 100 domain decomposition iterations in Table 4.2. For the OT algorithm, we have to differ between the subdomains, which are at the border of the image domain, i.e., the first and the last stripe in Figure 4.14(b), and the ones which are in the inner part of the image domain. The subdomains on the borders share only one interface with the neighbouring subdomain, resulting in only one η-iteration (3.8), while the other ones require the solution of two η-iterations on the lower and upper interfaces. Consequently, the solution of the subspace minimization problem with OT for the border elements is a bit faster than its solution for the inner elements. In either case, BOS-SB by far outperforms the OT algorithm in terms of computational speed. BOS-SB is about three times faster than OT in the computation of one subminimization problem.

Next we compare OT and BOS-SB in their ability to solve the domain decomposition problem accurately and fast. To do so, we first have to find a reasonable basis for comparison. In particular, we have to find the right stopping criterion. One standard choice for stopping an algorithm is to check the distance between two subsequent iterates, i.e., the value of $\|u^{(n+1)} - u^{(n)}\|_2$. If this value is smaller than a prescribed tolerance, i.e., if we are close to a fixed point of the algorithm, then the iteration is stopped and the current iterate is accepted as a good approximation to the minimizer. While this is a good criterion for stopping an algorithm, it is not a good one for comparing two algorithms with each other. The iterative behaviour of two algorithms can be very different and being close to a fixed point does not necessarily mean that the algorithm is close to the desired solution. The next generic choice then is the value of the energy evaluated in the iterates. But again, the energy value does not seem to be applicable too, because, although the energies decrease,

	OT	BOS-SB
$\mathcal{N} = 2$: ø CPU time / iteration	0.6 s	0.2 s
$\mathcal{N} = 3$: ø CPU time / iteration	0.52 s (border), 0.74 s (inner)	0.17 s
$\mathcal{N} = 4$: ø CPU time / iteration	0.46 s (border), 0.72 s (inner)	0.14 s
$\mathcal{N} = 5$: ø CPU time / iteration	0.46 s (border), 0.72 s (inner)	0.14 s

Table 4.2: *Computational performance of the* subminimization solvers *in the sequential version of the domain decomposition algorithm in* (4.36) *with* $\mathcal{N}$ *subdomains for inpainting of Figure 4.14(a): CPU times are compared for the OT strategy and the proposed Bregman Operator Splitting - Split Bregman strategy (BOS-SB) Algorithm 4. The reported CPU time is the time in seconds needed for one subspace minimization, averaged over the subdomains and over the first* 100 *domain decomposition iterations. For the evaluation of the OT algorithm one has to defer between a subminimization problem on the border of the image domain (only one interface and hence only one* η*-iteration) and a subminimization problem in the inner of the image domain (two interfaces and hence two* η*-iterations).*

the energy values seem to be different for the two algorithm, cf. Figure 4.16.

For image inpainting one quality measure is, how close the inpainted image is to the original image. While the original image is usually unknown in practice, for the comparative tests we are running, it seems to be a good measure for comparing the computational time of the two algorithms to reach a certain qualitative result. More precisely, let $e^{(n)} = \|u_{org} - u^{(n)}\|_2 / \|u^{(n)}\|_2$, $n = 1, 2, \dots$ be the error between the current iterate $u^{(n)}$ and the original image u_{org}. The algorithms are stopped when $e^{(n)}$ falls below a certain tolerance tol the first time. We stick to this choice for a quality measure and a stopping criterion for the two algorithms, although we again have to adapt the target tolerance to the two algorithms, cf. Figure 4.17.

Moreover, in Figure 4.18 we check the respective inpainting results in detail for the chosen tolerances $tol_1 = 0.02662$ for OT domain decomposition and $tol_2 = 0.0225$ for BOS-SB domain decomposition. The inpainting results seem to be comparable for these choices of stopping criteria.

From our numerical discussion up to now we have seen the BOS-SB is three times faster than OT in each subminimization problem, cf. Table 4.2, but that BOS-SB needs a larger number of iterations to achieve the same quality in the inpainting result as OT. What is the effect of these two subminimization strategies onto the performance of the domain decomposition algorithm as a whole? Looking at Figure 4.19 and Table 4.3 one immediately sees that the domain decomposition algorithm with BOS-SB is faster than OT, where the computational advantage of BOS-SB increases with the number of subdomains. In particular, a surprising result for us

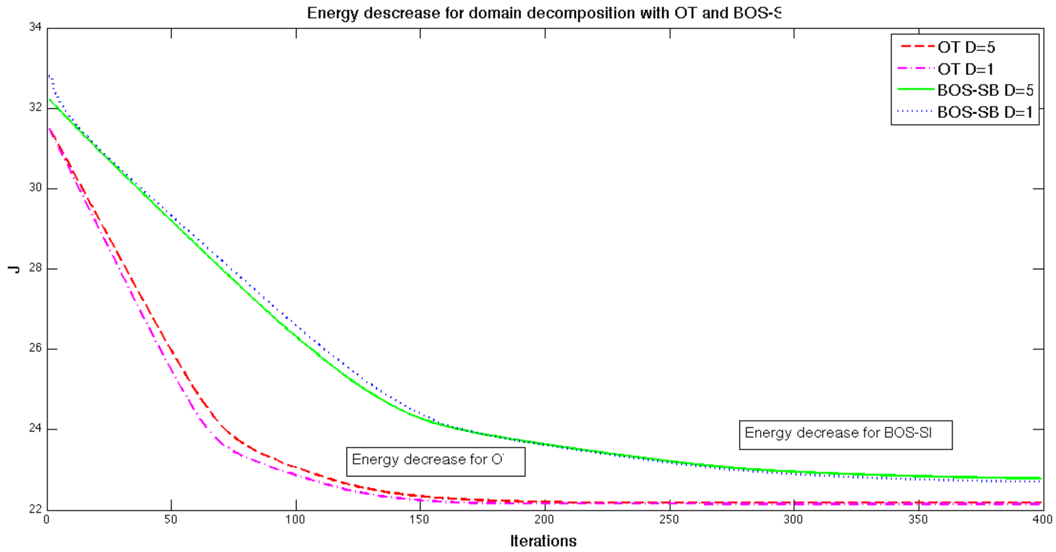
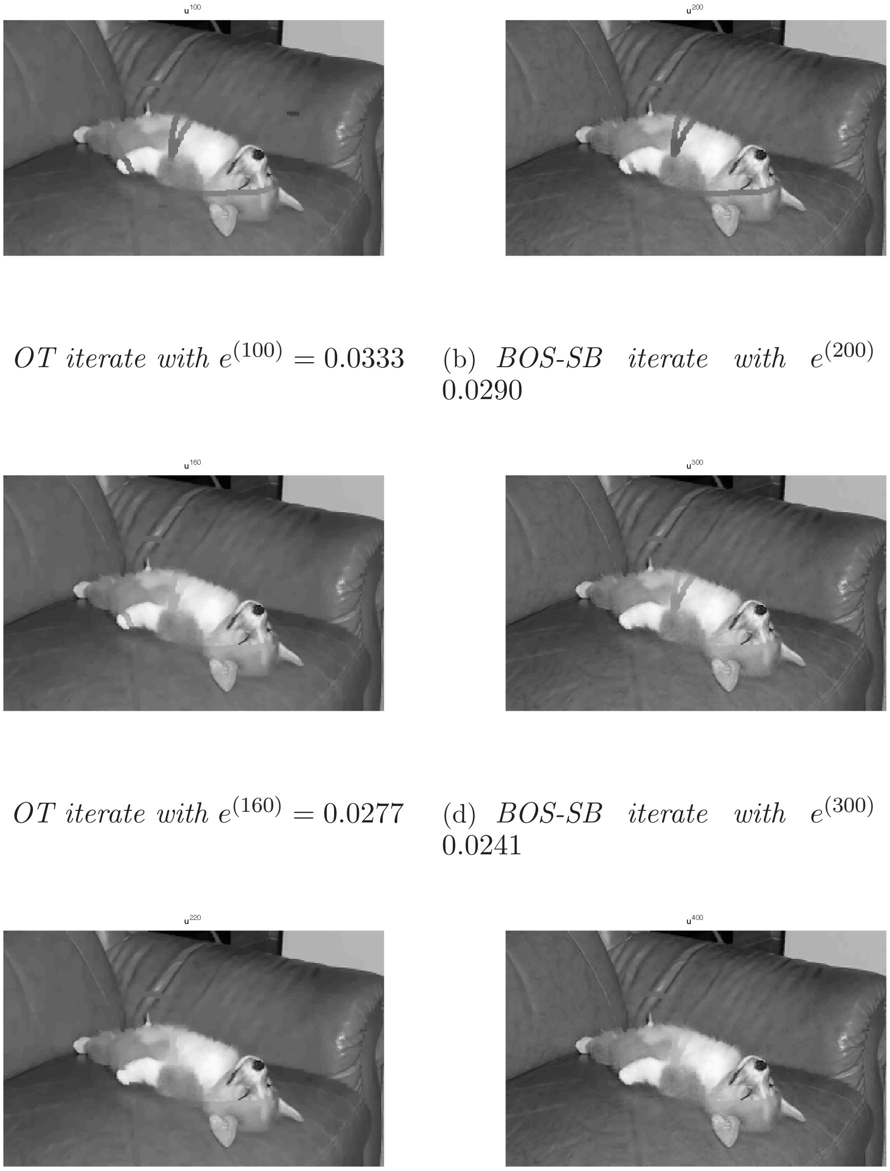


Figure 4.16: *Decrease of the energy $\mathcal{J}$ (4.1) for the OT algorithm and BOS-SB iteration, shown for the iterative minimization of $\mathcal{J}$ on the whole domain and the domain decomposition iteration with $\mathcal{N} = 5$ subdomains.*

was, that the CPU time for the sequential version of the algorithm computed with BOS-SB is - in contrast to the same computation with OT - only slightly increasing. Note, that we have not parallelized our computations yet.

Numerical results - parallel algorithm Finally, we also compare the parallel performance of the domain decomposition algorithm when computed with OT and BOS-SB. Note, that for the sequential version it turned out that solving the subminimization problems (4.47) with $sub = 1$ or $sub = 4$ BOS-SB computations does not make a significant difference in terms of computational time needed to solve the inpainting task with the domain decomposition algorithm, cf. Table 4.4. We take advantage of this fact for the parallel computations. Here, choosing $sub = 4$ and making more progress in each domain decomposition iteration with approximately the same computational effort, reduces the computational time as a whole because we reduce the number of domain decomposition iterations and hence, the amount of communication we have to do between the processes. As already discussed in [66], this strategy cannot be applied for the domain decomposition algorithm solved with OT. For the parallel computations we therefore choose $sub = 1$ for the algorithm with OT and $sub = 4$ for the algorithm solved with BOS-SB. See Table 4.5 and Figure 4.20 for the computational results for inpainting of the image in Figure 4.14(a).

We also test the parallel BOS-SB domain decomposition algorithm for a vandalized image of size 1768 × 2656 pixels, see Figure 4.21, where we decomposed it into



(a) *OT iterate with* $e^{(100)} = 0.0333$

(b) *BOS-SB iterate with* $e^{(200)} = 0.0290$

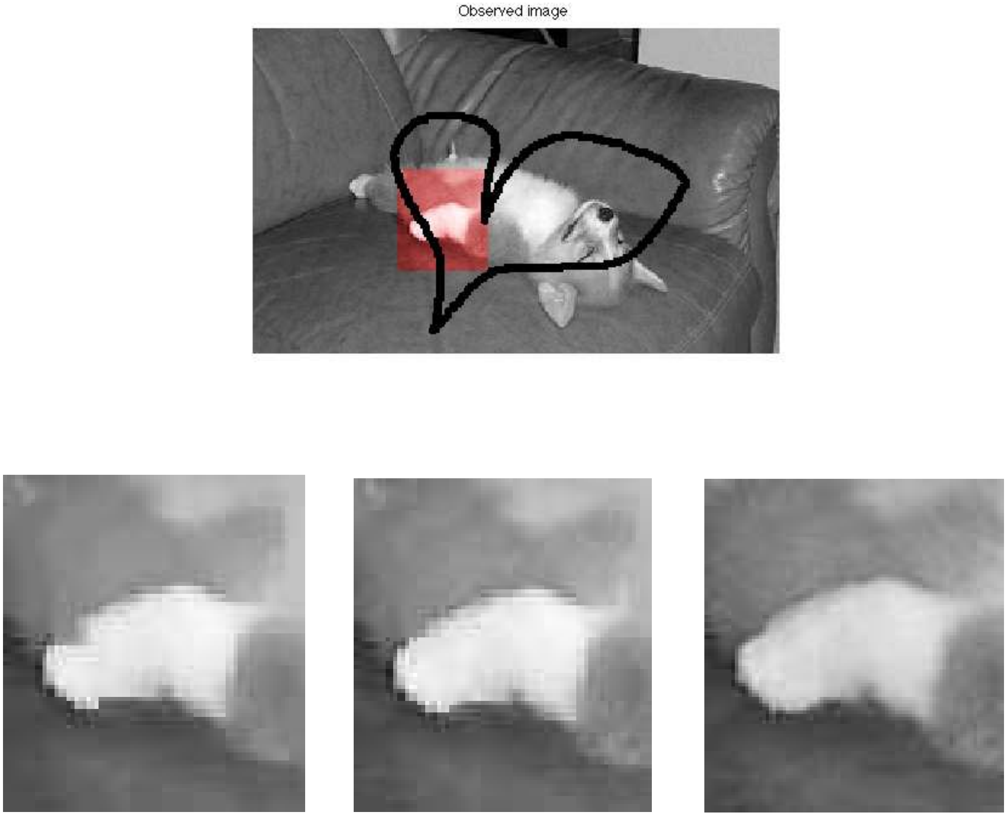
(c) *OT iterate with* $e^{(160)} = 0.0277$

(d) *BOS-SB iterate with* $e^{(300)} = 0.0241$

(e) *OT iterate with* $e^{(220)} = 0.02663$

(f) *BOS-SB iterate with* $e^{(400)} = 0.0224$

Figure 4.17: *Intermediate results of the inpainting result in Figure 4.15. The error* $e^{(n)}$ *in the* sequential domain decomposition algorithm (4.36) *evolves differently when computed with OT and when computed with BOS-SB.*



(a) OT with $tol_1 = 0.0225$

(b) BOS-SB with $tol_2 = 0.0225$

(c) Original image

Figure 4.18: *Detail of the inpainting result in Figure 4.15 and the original image. The* sequential domain decomposition algorithm (4.36) *with OT stops after* 620 *iterations with an error* $e^{(620)} = 0.02662$*, while the same algorithm with BOS-SB used for solving the subminimization problems terminates after* 406 *iterations with an error of* $e^{(406)} = 0.0225$*. Again the error* $e^{(n)} = \|u_{org} - u^{(n)}\|_2 / \|u^{(n)}\|_2$.

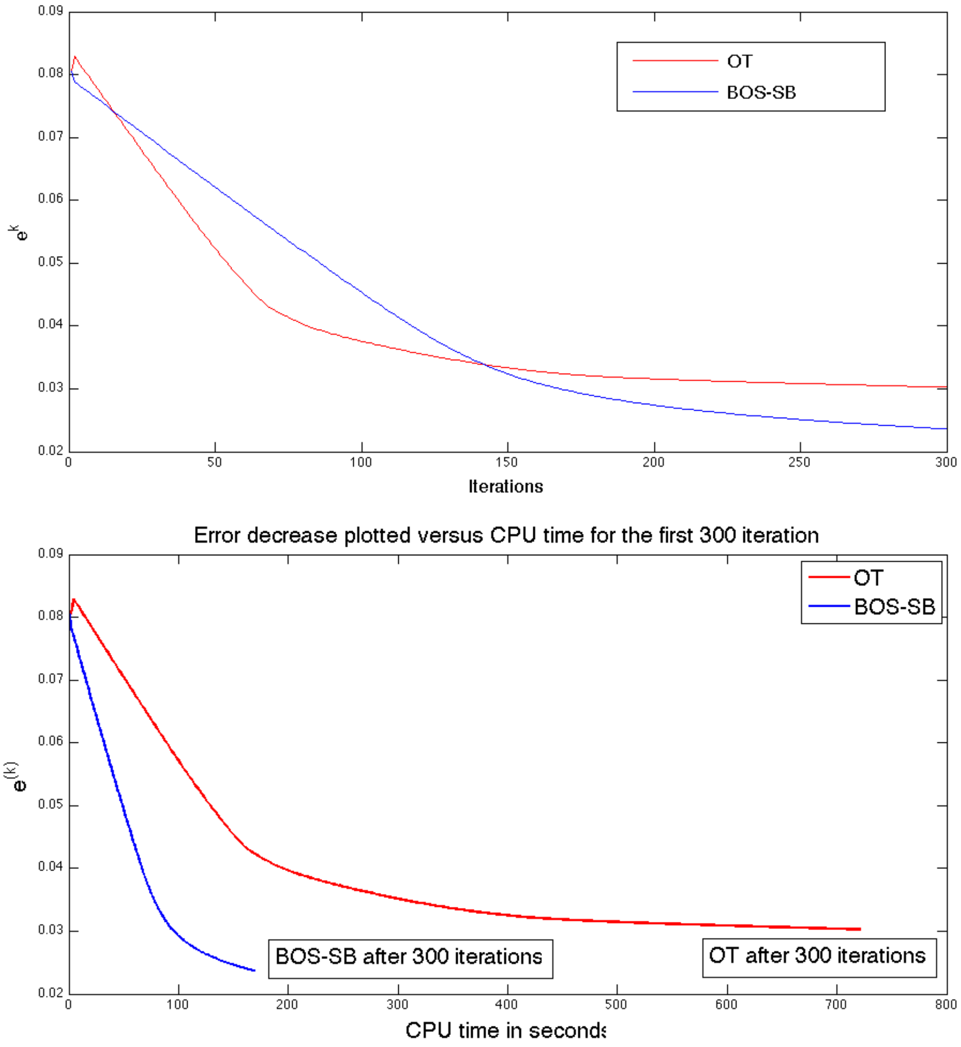


Figure 4.19: *Error decrease for inpainting of Figure 4.14(a) with the* sequential version *of the domain decomposition* (4.36) *in* $\mathcal{N} = 4$ *subdomains: in each domain decomposition iteration we measure the error between the original and the inpainted image, i.e.,* $e^{(n)} = \|u_{org} - u^{(n)}\|_2 / \|u^{(n)}\|_2$ *for iterations* $n = 1, 2, \dots 372$. *While the OT error decreases much faster at the beginning than the error in BOS-SB, it slows down as iterations progress and BOS-SB catches up.*

# domains	OT	BOS-SB (sub= 1)
$\mathcal{N} = 2$	223 iterations / 263.32 CPU s	399 iterations / 192.02 CPU s
$\mathcal{N} = 3$	290 iterations / 510.45 CPU s	405 iterations / 210.38 CPU s
$\mathcal{N} = 4$	843 iterations / 2004.82 CPU s	372 iterations / 210 CPU s
$\mathcal{N} = 5$	224 iterations / 636.65 CPU s	394 iterations / 242.39 CPU s
$\mathcal{N} = 6$	245 iterations / 817.45 CPU s	375 iterations / 251.62 CPU s

Table 4.3: *Inpainting for Figure 4.14(a): comparison of computational performance for the* sequential version *of the domain decomposition algorithm in* (4.36) *for using iterative thresholding versus BOS-SB to solve the subminimization problems. The domain decomposition algorithm with OT has been run until* $e^{(n)} < tol_1 = 0.02662$, *the domain decomposition algorithm with BOS-SB terminated when* $e^{(n)} < tol_2 = 0.0225$. *When increasing the number of subdomains* $\mathcal{N}$, *the CPU time seem to increase tremendously for OT, while the computational time for BOS-SB only slightly increases.*

# domains	BOS-SB (sub= 1)	BOS-SB (sub=4)
$\mathcal{N} = 2$	399 iterations / 192.02 CPU s	102×4 iterations / 191.18 CPU s
$\mathcal{N} = 3$	405 iterations / 210.38 CPU s	103×4 iterations / 206.8 CPU s
$\mathcal{N} = 4$	372 iterations / 210 CPU s	95×4 iterations / 208.69 CPU s
$\mathcal{N} = 5$	394 iterations / 242.39 CPU s	101×4 iterations / 245.54 CPU s
$\mathcal{N} = 6$	375 iterations / 251.62 CPU s	95×4 iterations / 247.86 CPU s

Table 4.4: *Inpainting for Figure 4.14(a): comparison of computational performance for the* sequential version *of the domain decomposition algorithm in* (4.36) *solved with BOS-SB with sub* = 1 *subminimization iteration* (4.47) *and with sub* = 4 *subminimization iterations.*

# domains	OT	BOS-SB
$\mathcal{N} = 4$	161 iterations / 135.87 CPU s *(reached accuracy* $e^{(k)} = 0.029$*)*	92×4 iterations / 65.95 CPU s *(reached accuracy* $e^{(k)} = 0.025$*)*

Table 4.5: *Inpainting for Figure 4.14(a): comparison of computational performance for the* parallel version *of the domain decomposition algorithm in* (4.40) *for using OT versus BOS-SB to solve the subminimization problems.*

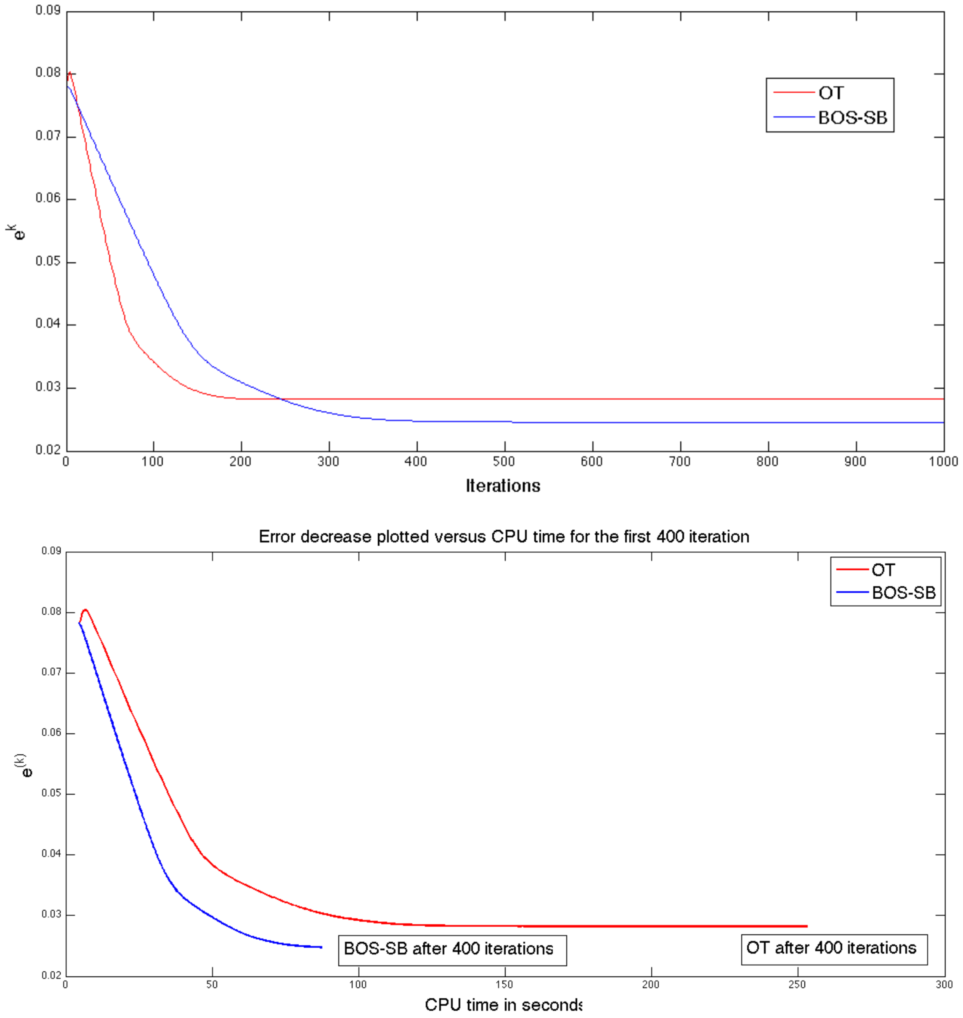


Figure 4.20: *Error decrease for inpainting of Figure 4.14(a) with the parallel version of domain decomposition* (4.40) *in four subdomains: in each domain decomposition iteration we measure the error between the original and the inpainted image, i.e.,* $e^{(n)} = \|u_{org} - u^{(n)}\|_2 / \|u^{(n)}\|_2$ *for iterations* $n = 1, 2, \ldots 1000$*. The final error of OT is* $e^{(1000)} = 0.028$*, while the error in BOS-SB in the final iteration is* $e^{(1000)} = 0.0246$*.*

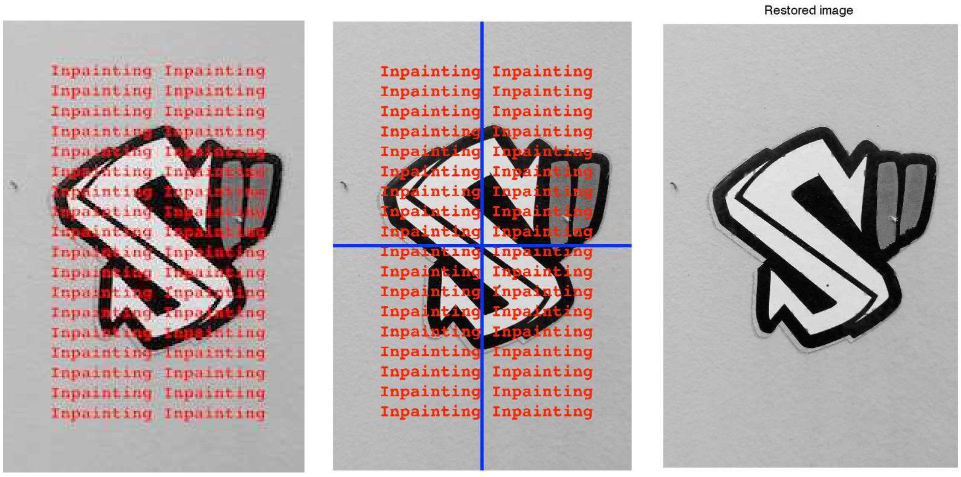


Figure 4.21: *Inpainting with the parallel domain decomposition strategy* (4.40)*: (l.) the vandalized image of size* 1768 × 2656 *pixels; (m.) its decomposition into four domains; (r.) the restored image computed with* (4.40)

# domains	BOS-SB
$\mathcal{N} = 1$	279 iterations / $\approx$ 15.8 CPU h
$\mathcal{N} = 4$	192 × 4 iterations / $\approx$ 3.26 CPU h
$\mathcal{N} = 16$	172 × 4 iterations / $\approx$ 2.2 CPU h

Table 4.6: *Inpainting for the* 1768×2656 *image in Figure 4.21: computational performance for the parallel version of the domain decomposition algorithm in* (4.40) *when using BOS-SB to solve the subminimization problems.*

four non-overlapping domains in order to restore it on multiple processors. A comparison with respect to the computation time for different numbers of subdomains is shown in Table 4.6. There we see that by increasing the number of domains, the number of iterations and the CPU time decrease.

Bibliography

[1] V.I. Agoshkov and V.I. Lebedev, *Poincare-Steklov operators and the methods of partition of the domain in variational problems*, In G.I. Marchuk (ed.), Computational Processes and Systems, no. 2, Nauka, Moscow, 1985, pp. 173–227 [Russian].

[2] L. Ambrosio, N. Fusco, and D. Pallara, *Functions of bounded variation and free discontinuity problems.*, Oxford Mathematical Monographs. Oxford: Clarendon Press. xviii, 2000.

[3] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing. Partial Differential Equations and the Calculus of Variation*, Springer, 2002.

[4] W. Baatz, M. Fornasier, P. Markowich, C.-B. Schönlieb, *Inpainting of ancient Austrian frescoes*, Conference proceedings of Bridges 2008, Leeuwarden, pp. 150–156, 2008.

[5] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, *Filling-in by joint interpolation of vector fields and grey levels*, IEEE Transactions on Image Processing, vol. 10, no. 8, pp. 1200–1211, 2001.

[6] V. Barbu and T. Precupanu, *Convexity and Optimization in Banach Spaces*, 1996.

[7] H. H. Bauschke, J. M Borwein, and A. S. Lewis, *The method of cyclic projections for closed convex sets in Hilbert space. Recent developments in optimization theory and nonlinear analysis*, (Jerusalem, 1995), 1–38, Contemp. Math., 204, Amer. Math. Soc., Providence, RI, 1997

[8] A. Beck and M. Teboulle, *Fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sciences **2**, 2009, pp. 183–202

[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image Inpainting*, In Computer Graphics Proceedings of SIGGRAPH 2000, 2000

[10] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, *Simultaneous structure and texture image inpainting*, UCLA CAM Report 02-47, July 2002.

[11] J.-F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu, *Variational formulation and algorithm for trace operator in domain decomposition calculations*, In T.F. Chan et al. (eds.), Domain Decomposition Methods, SIAM, Philadelphia, 1989, pp. 3–16

[12] A. Braides, Γ-*Convergence for Beginners*, No.22 in Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2002.

[13] K. Bredies and D. Lorenz, *Linear convergence of iterative soft-thresholding*, J. Fourier Anal. Appl. **14**(5-6), 2008, pp. 813-837.

[14] L. Bregman, *A relaxation method of finding a common point of convex sets and its application to the solution of convex programming problems.* J. Wych. Math. and Math.Phys. **7**(620-631), 1967, pp. 200–217

[15] R. E. Bruck and S. Reich, *Nonexpansive projections and resolvents of accretive operators in Banach spaces.* Houston J. Math. **3**(4), 1977, pp. 459–470

[16] E. J. Candès, *Compressive sampling*, Int. Congress of Mathematics, **3**, pp. 1433-1452, Madrid, Spain, 2006

[17] E. J. Candès, J. Romberg, and T. Tao, *Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inf. Theory **52**(2), 2006, pp. 489–509.

[18] E. J. Candès and T. Tao, *Near Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?*, IEEE Trans. Inf. Theory **52**(12), 2006, pp. 5406–5425.

[19] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman Iterations for compressed sensing*, Math. Comp. **78**(267), 2009, pp. 1515–1536

[20] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the Linearized Bregman Iteration for ℓ_1-norm minimization*, Math. Comp. **78**, 2009, pp. 2127–2136

[21] C. Carstensen, *Domain decomposition for a non-smooth convex minimalization problems and its application to plasticity*, Numerical Linear Algebra with Applications **4**(3), 1998, pp. 177–190.

[22] E. Casas, K. Kunisch, and C. Pola, *Some Applications of BV Functions in Optimal Control and Calculs of Variations*, ESAIM: Proceedings Contrôle et Équations aux Dérivées Partielles **4**, 1998, pp. 83–96

[23] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision **20**(1-2), 2004, pp. 89–97.

[24] A. Chambolle, *An algorithm for Mean Curvature Motion*, Interfaces and Free Boundaries **6**, 2004, pp. 195–218.

[25] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, *An Introduction to Total Variation for Image Analysis.* In M. Fornasier (ed.), Theoretical Foundations and Numerical Methods for Sparse Recovery, Radon Series on Computational and Applied Mathematics, De Gruyter Verlag, 2010, pp. 263-340.

[26] A. Chambolle, J. Darbon, *On total variation minimization and surface evolution using parametric maximum flows*, Int. J. Comput. Vis. **84**, 2009, 288–307

[27] A. Chambolle and P.-L. Lions, *Image recovery via total variation minimization and related problems.*, Numer. Math. **76**(2), 1997, pp. 167–188.

[28] R. H. Chan, C.-W. Ho, and M. Nikolova, *Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Details-Preserving Regularization*, IEEE Transactions on Image Processing **14**(10), 2005, pp. 1479–1485.

[29] T. F. Chan, G. H. Golub, and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput. **20**(6), 1999, pp. 1964–1977.

[30] T. F. Chan, and T. P. Mathew, Domain decomposition algorithms, Acta Numerica **3**, 1994, pp. 61–143.

[31] T. F. Chan and J. Shen, *Mathematical models for local non-texture inpaintings*, SIAM J. Appl. Math., 62(3),2001, pp. 1019–1043.

[32] T. F. Chan and J. Shen, *Nontexture inpainting by curvature driven diffusions (CDD)*, J. Visual Comm. Image Rep. **12**(4), 2001, pp. 436449.

[33] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005

[34] P. G. Ciarlet, *The Finite Element Methods for Elliptic Problems*, Classics Appl. Math. 40, SIAM, Philadelphia, 2002

[35] J. P. Cocquerez, L. Chanas, and J. Blanc-Talon, *Simultaneous Inpainting and Motion Estimation of Highly Degraded Video-Sequences*, Preprint, 2001.

[36] A. Cohen, *Numerical Analysis of Wavelet Methods.*, Studies in Mathematics and its Applications 32. Amsterdam: North-Holland., 2003.

[37] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore, *Harmonic Analysis of the space BV*, Rev. Mat. Iberoamericana **19**, 2003, pp. 235–263.

[38] A. Cohen, R. DeVore, P. Petrushev, and H. Xu, *Nonlinear Approximation and the space $BV(\mathbb{R}^2)$*, Amer. J. Math. **121**, 1999, pp. 587–628.

[39] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul. **4**(4), 2005, pp. 1168–1200.

[40] G. David, *Global Minimizers of the Mumford-Shah Functional*, Current developments in mathematics, 1997 (Cambridge, MA), Int. Press, Boston, MA, 1999, pp. 219–224.

[41] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.

[42] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems*, Comm. Pure Appl. Math. **57**(11), 2004, pp. 1413–1457. Regularization

[43] I. Daubechies, R. DeVore, M. Fornasier, and S. Güntürk, *Iteratively reweighted least squares minimization for sparse recovery*, Commun. Pure Appl. Math. **63**(1), 2010, pp. 1-38.

[44] I. Daubechies, M. Fornasier, and I. Loris, *Acceleration of the projected gradient method for linear inverse problems with sparsity constraints*, J. Fourier Anal. Appl. **14**(5-6), 2008, pp. 764–792.

[45] I. Daubechies, G. Teschke, and L. Vese, *Iteratively solving linear inverse problems under general convex constraints*, Inverse Probl. Imaging **1**(1), 2007, pp. 29–46.

[46] J. Darbon and M. Sigelle, *A fast and exact algorithm for total variation minimization*, IbPRIA 2005 **3522**(1), 2005, pp. 351-359.

[47] J. Darbon and M. Sigelle, *Image Restoration with Discrete Constrained Total Variation Part I: Fast and Exact Optimization*, J. Math. Imaging and Vision **26**(3), 2006, pp. 261-276.

[48] G. Dal Maso, *An Introduction to* Γ*-Convergence*, Birkhäuser, Boston, 1993.

[49] E. De Giorgi, *Convergence problems for functionals and operators*, Recent methods in nonlinear analysis, Proc. Int. Meet., Rome 1978, 1979, pp. 131-188.

[50] E. De Giorgi and T. Franzoni, *Su un tipo di convergenza variazionale*, Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur. **68**, 1975, pp. 842-850.

[51] D. Dobson and C. R. Vogel, *Convergence of an iterative method for total variation denoising*, SIAM J. Numer. Anal. **34**(5), 1997, pp. 1779–1791.

[52] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inf. Theory **52**(4), 2006, pp. 1289–1306.

[53] I. Ekeland and R. Temam, *Convex analysis and variational problems. Translated by Minerva Translations, Ltd., London.*, Studies in Mathematics and its Applications. Vol. 1. Amsterdam - Oxford: North-Holland Publishing Company; New York: American Elsevier Publishing Company, Inc., 1976.

[54] G. Emile-Male, *The Restorer's Handbook of Easel Painting*, Van Nostrand Reinhold, New York, 1976

[55] H.W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems.*, Mathematics and its Applications (Dordrecht). 375. Dordrecht: Kluwer Academic Publishers., 1996.

[56] S. Esedoglu and J. Shen, *Digital image inpainting by the Mumford - Shah - Euler image model*, European J. Appl. Math. **13** , 2002, pp. 353370.

[57] L. C. Evans and R. F. Gariepy, *Measure Theory and Fine Properties of Functions.*, CRC Press, 1992.

[58] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, Special Issue on Convex Optimization Methods for Signal Processing **1**(4), 2007, pp. 586–598.

[59] K. Frick and O. Scherzer, *Regularization of Ill-Posed Linear Equations by the Non-stationary Augmented Lagrangian Method*, J. Integral Equations Appl. **22**(2), 2010, pp. 217–257.

[60] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*, Wiley, New York, 1999

[61] M. Fornasier, *Domain decomposition methods for linear inverse problems with sparsity constraints*, Inverse Problems **23**, 2007, pp. 2505–2526.

[62] M. Fornasier, *Numerical Methods for Sparse Recovery*, Theoretical Foundations and Numerical Methods for Sparse Recovery (M. Fornasier, ed.), Radon Series on Computational and Applied Mathematics, De Gruyter Verlag, 2010.

[63] M. Fornasier,Y. Kim, A. Langer, and C.-B. Schönlieb, *Wavelet Decomposition Method for L_2/TV-Image Deblurring*, 2011, preprint.

[64] M. Fornasier, A. Langer, and C.-B. Schönlieb, *A convergent overlapping domain decomposition method for total variation minimization*, Numerische Mathematik **116**(4), 2010, pp. 645–685.

[65] M. Fornasier and R. March, *Existence of minimizers of the Mumford and Shah functional with singular operators in two space dimensions*, submitted to SIAM J. Math. Anal., March 2010, pp. 27.

[66] M. Fornasier and C.-B. Schönlieb, *Subspace correction methods for total variation and ℓ_1-minimization*, SIAM J. Numer. Anal. **47**(5), 2009, pp. 3397–3428

[67] M. Fornasier and R. Ward, *Iterative thresholding meets free-discontinuity problems*, Found. Comput. Math. **10**(5), 2010, pp. 527–567.

[68] S. Geman and D. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Retoration of Images*, IEEE Trans. PAMI PAMI-6, 1984, pp. 721–741.

[69] T. Goldstein, S. Osher, *The split Bregman method for L^1 regularized problems*, UCLA CAM Report 08-29, 2008.

[70] M. Grassmair, M. Haltmeier, and O. Scherzer, *Sparse regularization with ℓ^q penalty term*, Inverse Problems **24**, 2008, pp. 113.

[71] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I*, Vol. 305 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag: Berlin, 1996

[72] K. Ito and K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, Series: Advances in Design and Control (No. 15) SIAM, 2008

[73] S.-H. Kang, T.-F. Chan, and S. Soatto, *Landmark based inpainting from multiple views*, UCLA CAM Report 02-11, March 2002.

[74] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *A Method for ℓ_1-regularized least squares*, IEEE Journal of Selected Topics in Signal Processing **1**(4), 2007, pp. 606–617

[75] A. Langer, *Convergence Analysis of a Nonoverlapping Domain Decomposition Algorithm for Total Variation Minimization*, 2009, preprint.

[76] A. Langer, S. Osher, and M. Fornasier, *Bregmanized Domain Decomposition for Image Restoration*, submitted to Journal of Scientific Computing, pp. 25.

[77] P.-L. Lions, *On the Schwarz alternating method I.* In R. Glowinski, G.H. Golub, G.A. Meurant, J. Périaux (eds.), First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1988, pp. 1–42.

[78] P.-L. Lions, *On the Schwarz Alternating Method II: Stochastic Interpretation and Order Properties.* In: T.F. Chan, R. Glowinski, J. Priaux, O.B. Widlund (eds.), Proceedings of the 2nd International Conference on Domain Decomposition Methods, SIAM, Philadelphia, PA, 1989, pp. 47–70.

[79] P.-L. Lions, *On the Schwarz Alternating Method III: a variant for non-overlapping subdomains.* In T.F. Chan et al. (eds), Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, pp. 202–231.

[80] I. Loris, *On the performance of algorithms for the minimization of* 1*-penalized functionals*, Inverse Problems **25**, 2009.

[81] M. Lustig, D. Donoho, and J. M. Pauly, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magnetic Resonance in Medicine **58**(6), 2007, pp. 1182–1195.

[82] F. Maddalena and S. Solimini, *Lower semicontinuity properties of functionals with free discontinuities*, Arch. Ration. Mech. Anal. **159**(4), 2001, pp. 273–294.

[83] F. Malgouyres, *Increase in the resolution of digital images: Variational theory and applications.* Doctoral dissertation, Ecole Normale Suprieure de Cachan, Cachan, France, 2000.

[84] F. Malgouyres and F. Guichard, *Edge direction preserving image zooming: A mathematical and numerical analysis*, SIAM J. Numer. Anal. **39**(1), 2001, pp. 137.

[85] L. D. Marini and A. Quarteroni, *A relaxation procedure for domain decomposition methods using finite elements*, Numer. Math. **55**, 1989, pp. 575–598

[86] S. Masnou and J.-M. Morel, *Level-lines based disocclusion*, Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP-98), Chicago, Illinois, October 4-7, 1998, vol. 3, pp. 259263. IEEE Computer Society, 1998.

[87] J.-M. Morel and S. Solimini, *Variational Methods in Image Segmentation*, Birkhäuser, Boston, 1995.

[88] J. Müller, *Parallel Methods for Nonlinear Imaging Techniques*, Master thesis, University of Münster, 2008.

[89] D. Mumford and J. Shah, *Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems*, Comm. Pure Appl. Math. **42**, 1989, pp. 577–685.

[90] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence* $\mathcal{O}(\frac{1}{k^2})$, Doklady AN SSSR (translated as Soviet Math. Docl.) **269**, 1983, pp. 543–547.

[91] Y. Nesterov, *Introductory Lectures on Convex Optimization: A basic course*, Kluwer, Boston, 2004

[92] Y. Nesterov, *Smooth minimization of non-smooth functions*. Mathematic Programming, Ser. A **103**, 2005, pp. 127–152.

[93] M. Nitzberg, D.Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*, Springer-Verlag, Lecture Notes in Computer Science, 662, Berlin-New York, 1993

[94] T. Ohtsuka, K. Shirakawa, and N. Yamazaki, *Optimal control problem for Allen-Cahn type equation associated with total variation energy*, MIMS Technical Report No.00023 (200912221), 2009, pp. 28.

[95] Z. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings*, Bull. Amer. Math. Soc. **73**, 1967, pp. 591–597.

[96] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul. **4**(2), 2005, pp. 460–489.

[97] S. Osher, Y. Mao, B. Dong, W. Yin, *Fast Linearized Bregman Iteration for compressed sensing and sparse denoising*, Comm. Math. Sci. **8**(1), 2010, pp. 93–111.

[98] T. Pock, M. Unger, D. Cremers, and H. Bischof, *Fast and Exact Solution of Total Variation Models on the GPU*, In Computer Vision and Pattern Recognition, 2008, pp. 1-8.

[99] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, *An algorithm for Minimizing the Mumford-Shah Functional*, In ICCV Proceedings, LNCS, Springer, 2009

[100] A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, Numerical Mathematics and Scientific Computation, The Clarendon Press Oxford University Press, Oxford Science Publications, New York, 1999.

[101] R.T. Rockafellar, *Convex Analysis*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, 1970.

[102] R.T. Rockafellar and R.J.B. Wets, *Variational analysis*, Grundlehren der Mathematischen Wissenschaften, vol. 317, Springer-Verlag, Berlin, 1998.

[103] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D **60**(1-4), 1992, pp. 259–268.

[104] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, F. Lenzen, *Variational Methods in Imaging*, Applied Mathematical Sciences, vol. 167, Springer, New York, 2009.

[105] C.-B. Schönlieb, *Modern PDE Techniques for Image Inpainting*, PhD-Thesis, University of Cambridge, 2009

[106] H.W. Schwarz, *Über einige Abbildungsaufgaben*, J. Reine Angew. Meth. **70**, pp. 105–120.

[107] S. Setzer, *Split Bregman Algorithm, Douglas-Rachford Splitting and Frame Shrinkage*. In X.-C. Tai et al. (ed.), Proceedings of the Second International Conference on Scale Space Methods and Proceedings of the 2nd International Conference on Scale Space and Variational Methods in Computer Vision, LNCS, vol. 5567, Springer, Berlin, 2009, pp. 464-476.

[108] X.-C. Tai and P. Tseng, *Convergence rate analysis of an asynchronous space decomposition method for convex minimization*, Math. Comp. **71**(239), 2001, pp. 1105–1135.

[109] X.-C. Tai and J. Xu, *Global convergence of subspace correction methods for convex optimization problems*, Math. Comp. **71**(237), 2002, pp. 105–124.

[110] G. Teschke and R. Ramlau, *An Iterative Algorithm for Nonlinear Inverse Problems with Joint Sparsity Constraints in Vector Valued Regimes and an Application to Color Image Inpainting*, Inverse Problems **23**, 2007, pp. 1851–1870.

[111] A. Toselli and O.Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Springer Series in Computational Mathematics, Springer, Berlin, 2005

[112] A. Tsai, J. A. Yezzi, and A. S. Willsky, *Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification*, IEEE Trans. Image Process. **10**(8), 2001, pp. 11691186.

[113] L. Vese, *A study in the BV space of a denoising-deblurring variational problem.*, Appl. Math. Optim. **44**, 2001, pp. 131–161.

[114] L. A. Vese and S. J. Osher, *Modeling textures with total variation minimization and oscillating patterns in image processing*, UCLA CAM Report 02-19, May 2002.

[115] C.R. Vogel, M.E. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput. **17**(1), 1996, pp. 227–238

[116] C. Vonesch and M. Unser, *A Fast multilevel algorithm for wavelet-regularized image restoration*, IEEE Transactions on Image Processing **18**(3), 2009, pp. 509–523.

[117] S. Walden, *The Ravished Image*, St. Martin's Press, New York, 1985

[118] J. Warga, *Minimizing certain convex functions*, J. Soc. Indust. Appl. Math. **11**, 1963, pp. 588–593.

[119] P. Weiss, L. Blanc-Féraud, and G. Aubert, *Efficient schemes for total variation minimization under constraints in image processing*, SIAM J. Sci. Comput. **31**(3), 2009, pp. 2047-2080.

[120] J. Xu, *The method of subspace corrections*, J. Comp. Appl. Math. 128, 2001, pp. 335–362

[121] J. Xu, X.-C. Tai and L.-L. Wang, *A Two-level domain decomposition method for image restoration*, UCLA Computational and Applied Mathematics Report 09-92, November 2009.

[122] W. Yin, *Analysis and generalization of the linearized Bregman method*, Rice University CAAM Technical Report TR09-02, 2009

[123] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imag. Sci. **1**(1), 2008, pp. 143–168.

[124] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction*, SIAM J. Imag. Sci. **3**(3), 2010, pp. 253–276.

[125] M. Zhu and T. Chan, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, UCLA, Center for Applied Math., CAM Reports 08-34, 2008.

[126] Matlab code and numerical experiments of the overlapping and non-overlapping domain decomposition methods provided in this thesis can be downloaded at the web-page: `http://homepage.univie.ac.at/carola.schoenlieb/webpage_tvdode/tv_dode_numerics.htm`

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, Juli 2011

Andreas Langer

Curriculum Vitae

Name:	Andreas Langer
Citizenship:	Austrian
Date of Birth:	April 1 1981

EDUCATION

Feb. 2008 - present	Ph.D. student at Johannes Kepler Universität Linz, Austria
Sept. 2006	Master degree in Technichal Mathematics (with distinction), Johannes Kepler Universität Linz, Austria
Oct. 2000 - Sept. 2006	Studies in Technical Mathematics, Johannes Kepler Universität Linz, Austria
June 1999	High School Diploma (Matura), Gymnasium und Realgymnasium des Schulvereines Kollegium Aloisianum, Linz, Austria

POSITIONS

Feb. 2008 - present	Research assistant at Johann Radon Institute for Computational and Applied Mathematics, Group “Analysis of Partial Differential Equations”, Linz, Austria
July 2007 - Jan. 2008	Employee at Raiffeisen Landesbank Oberösterreich / Treasury Middle Office, Linz, Austria
Nov. 2006 - June 2007	Research assistant at the Institute for Analysis/Group Dynamical Systems and Approximation Theory, Johannes Kepler Universität Linz, Austria

OTHER INVITED RESEARCH VISITS

Jan. 24 - Jan. 29 2011	Department of Applied Mathematics and Theoretical Physics (DAMPT), University of Cambridge, United Kingdom

March 22 - March 26 2010	Institute of Numerical and Applied Mathematics, Georg-August Universität, Göttingen, Germany
Sept. 15 - Dec. 15 2009	Department of Mathematics, University of California, Los Angeles (UCLA), U.S.A.

PUBLICATIONS

Submitted preprints to refereed journals

1. Langer, A., Osher, S., and Schönlieb, C.-B.: *Bregmanized Domain Decomposition Methods for Image Restoration*, submitted to J. Scientific Computing, May 2011, 25 pp.

2. Fornasier, M., Kim, Y., Langer, A., and Schönlieb, C.-B.: *Wavelet Decomposition Method for L2/TV-Image Deblurring*, submitted to SIAM J. Imaging Sciences, Jan 2011, 19 pp.

Refreed journal papers

3. Fornasier, M., Langer, A., and Schönlieb, C.-B.: *A Convergent Overlapping Domain Decomposition Method for Total Variation Minimization*, Numer. Math., Vol 116, No 4, 2010, pp. 645-685

Conference papers

4. Fornasier, M., Langer, A., and Schönlieb, C.-B.: *Domain Decomposition Methods for Compressed Sensing*, Proc. Int. Conf. SampTA09, Marseilles, 2009. arXiv:0902.0124v1 [math.NA]

SOFTWARE

Matlab code: Overlapping domain decomposition methods for total variation minimization

ORAL PRESENTATIONS

Contributed conference presentations

1. The Adaptive Iterative Bregman Algorithm, GAMM 2011 (82nd Annual Meeting of the International Association of Applied Mathematics and Mechanics). Graz (Austria), April 18-21, 2011

2. Domain decomposition methods for total variation minimization, Workshop on Numerical Methods for Optimal Control and Inverse Problems (OCIP 2011). Garching by Munich (Germany), March 14-16, 2011

3. Subspace correction algorithms for ℓ_1-norm and total variation minimization, SIMAI Congress 2010. Cagliari (Italy), June 21-25, 2010

4. Domain decomposition methods for total variation minimization, Söllerhaus Workshop on Domain Decomposition Solvers for Heterogeneous Field Problems. Hirschegg (Austria), June 2-6, 2010

5. Subspace correction methods for ℓ_1-norm and total variation minimization, SIAM conference on Imaging Science. Chicago (U.S.A.), April 12-14, 2010

6. Domain decomposition methods for compressed sensing, 8th international conference on Sampling Theory and Applications (SAMPTA 2009). Marseille (France), May 18-22, 2009

Seminars at universities and research institutions

7. Domain decomposition methods for total variation minimization, Invited Seminar, Universität Wien, Wien (Austria), May 05, 2011

8. Domain decomposition methods and Bregman Iterations for Total Variation Minimization, Invited Seminar, University of Edinburgh, Edinburgh (U.K.), April 07, 2011

9. Domain decomposition for totalvariation minimization, Invited Seminar, Georg-August Universität, Göttingen (Germany), March 31, 2011

10. Domain decomposition for total variation minimization, Invited Seminar, Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS), Toulouse (France), March 28, 2011

11. Domain decomposition for total variation minimization, Invited Seminar, Karl-Franzens Universität, Graz (Austria), March 24, 2011

12. Analysis of the Adaptive Iterative Bregman Algorithm, Applied and Computational Analysis Graduate Seminar. University of Cambridge, Cambridge (U.K.), January 28, 2011

13. Domain decomposition methods for local and nonlocal total variation minimization, Group Seminar START Project and PDE, Johann Radon Institute for Computational and Applied Mathematics (RICAM), Linz (Austria), February 02, 2010

14. Wavelet space decomposition for ℓ_1-norm and total variation minimization, Seminar: Applied Mathematics. University of California, Los Angeles (UCLA), Los Angeles (U.S.A.), October 23, 2009

15. Subspace correction methods for ℓ_1-norm and total variation minimization, Applied Math Colloquium. University of California, Los Angeles (UCLA), Los Angeles (U.S.A.), October 14, 2009

16. A convergent domain decomposition method for total variation minimization, RICAM Informal mini-workshop on sparsity and computations. Johann Radon Institute of Computational and Applied Mathematics (RICAM), Linz (Austria), June 03, 2009

PROFESSIONAL ACTIVITIES

Journal Reviewing

Signal Processing

Conference, workshop, meeting organization

2009: Contribution to the organization of the Summer School on "Theoretical Foundations and Numerical Methods for Sparse Recovery" held at the Johann Radon Institute for Computational and Applied Mathematics (RICAM), Linz - Austria, on August 31 - September 4, 2009
`http://www.ricam.oeaw.ac.at/events/summerschool_2009/`