وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR –ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار عنابة

Faculté des Sciences

Département de Mathématiques

Année : 2016

THESE

Présentée en vue de l'obtention du diplôme
de DOCTORAT en Maths

Option
Mathématiques appliquées
Titre

Mathematical Techniques in Image Processing (Techniques Mathématiques dans le Traitement d'Images)

Par

ZEGHBIB FATIMA ZOHRA

DIRECTRICE DE THESE : M^me^ **Nouri Fatma Zohra Prof. U.B.M. Annaba**

Devant le jury

PRESIDENT:

Mr. Haiour Mohamed	Professeur	Univ.B.M. Annaba

EXAMINATEURS:

Mr. Daili Nouredine	Professeur	Univ. Setif 1
Mr. Taallah Frekh	M.C.A.	Univ.B.M. Annaba
Mr. Oueld Hamouda Amar	M.C.A.	E.N.S. Kouba-Alger
Mr. Heller Benjamin	Directeur de Recherche (Principle Research Fellow)	Sheffield Hallam University, UK

Remerciements

Je tiens tout d'abord à remercier très vivement ma directrice de thèse Mme. NOURI Fatma Zohra, Professeur de l'enseignement supérieur, d'avoir accepté de diriger ce travail de recherche. Elle m'a encouragé. Je lui exprime ma plus profonde reconnaissance de m'avoir accompagné pendant les années de préparation de cette thèse. Elle a toujours apporté sa rigueur scientifique à certains points clés de ce travail. Je lui suis particulièrement reconnaissant de me soutenir tout en m'accordant sa confiance avec une grande liberté d'action. Elle m'a aidé à progresser dans ma réflexion grâce à ses conseils, son esprit critique et sa disponibilité.

J'adresse mes sincères remerciements à tous les membres de jury pour l'intérêt qu'ils ont bien voulu porter à ce travail de thèse :

- *Monsieur Haiour Mohamed, Professeur à l'université Badji Mokhtar-Annaba pour l'honneur qui me fait de présider ce jury, je la remercie très vivement.*
- *Monsieur Daili Nourreddine, Professeur à l'université Setif 1*
- *Monsieur Taallah Frekh, Maître de conférences à l'université Badji Mokhtar-Annaba,*
- *Monsieur Ouled Hamouda Ammar Maître de conférences à ENS Kouba.*
- *Monsieur Ben Haller, Directeur de Recherche (Principle Research Fellow) au Centre for Sports Engineering Research, Sheffield Hallam University, uk*

D'avoir accepté d'examiner ce travail. Ceci fût pour moi un honneur, je les remercie chaleureusement pour leurs critiques constructives, les remarques pertinentes et les corrections qui ont amélioré la présentation de ce manuscrit.

Un grand merci à mon Maré, qui a sacrifié sa recherche scientifique en Mathématiques et a crée les conditions favorables pour me soutenir afin que je puisse finaliser ma thèse.

Enfin et non les moins remerciements également à ma grande famille, mes enfants, mes parents, mes frères et sœurs qui m'ont soutenu pendant toutes les périodes de mes études et surtout mon père « Cherif ».

Merci à tous ceux qui ont contribué de prés ou de loin pour la réalisation de ce travail.

TABLE OF CONTENTS

Abstract: In the last decades, many approaches and various algorithms are introduced to minimize the functional calculus of the Tixotrop model. These classical techniques are formulated in an iterative form, that will cost a lot of CPU time. To overcome this disadvantage, one can use the decomposition method which reduces the problem to a sequence sub-problems of a more manageable size. In this particular context, we introduce a method of correction subspace based on the decomposition methods for the minimization of the Tixotrop model. These methods allow us to divide the space of the initial problem into several subspaces, to solve sub-problems faster. Then the solution of the original problem is obtained from the sub-problems'ones . The main difficulty in the resolution the tixotrop model by a domain decomposition technique has a non regular behavior of the borders, with discontinuities preserving. As the non regular model of the tixotrop is not adapted for the decomposition algorithm with matching and non matching to minimize the functional associated to the problem, after regularization; we provide several tests showing the effectiveness of the algorithm for image restoration. Furthermore we present a comparison of these results with the same algorithm by the total variation method.

Keywords: Image Processing, Domain Decomposition, Tixotrop Model.

Résumé: Dans les dernières décennies, de nombreuses approches et différents algorithmes sont introduits pour minimiser la fonctionnelle du modèle Tixotrop. Ces techniques classiques sont formulées sous forme des suites itératives; mais ceci présente un inconveient dans le coût de calcul de computation durant la résolution. Pour surmonter cet inconvénient, on peut utiliser la méthode de décomposition du domaine qui permet de réduire le problème à une suite finie de sous problèmes d'une taille plus gérable.

Dans ce contexte particulier, nous introduisons une méthode de correction de sous-espaces, basée sur les méthodes de décomposition de domaines pour la minimisation du modèle Tixotrop. Ces méthodes permettent de diviser l'espace du problème initial en plusieurs sous-espaces plus petits ceci qui permet de résoudre les sous problèmes plus rapidement. Ensuite la solution du problème original est obtenue à partir des solutions des sous problèmes associés. La difficulté essentielle dans la résolution du modèle tixotrop par la méthode de décomposition du domaine est le comportement non régulier des frontières des correctifs, avec la préservation des discontinuités. Comme le modèle tixotrop est non régulier, on ne peut pas adapter l'algorithme de décomposition de domaines avec recouvrement et sans recouvrement pour minimiser la fonctionnelle associée au problème, après la régularisation. On donne plusieurs tests numériques, on montre l'efficacité de l'algorithme pour la restauration d'images. de plus on présente une comparison de ces résultats avec les résultats obtenus par le même algorithme par la méthode de la variation totale.

Mots-clés: Traitement d'Images, Décomposition de Domaines, Modèle Tixotrop.

ملخص

خلال السنوات الأخيرة، قدمت العديد من الطرق والخوارزميات المختلفة لحل مسألة التيكسوتروب 'Tixotrop ' غير أن هذه الطرق والخوارزميات ترتكز في البحث عن الحل بطريقة المتتاليات التراجعية و هذا يستغرق الكثير من الوقت . و لتفادي هذه المشكلة هناك طريقة تفكيك النطاق التي تقوم بتجزئة المسألة الأصلية إلى عدد منتهي من المسائل الجزئية المعرفة على نطاقات جزئية يمكن حلها بسهولة.

وفي هذا السياق نقوم باستعمال طريقة تصحيح الفضاءات الجزئية التي تعتمد على طريقة تفكيك النطاق لحل نموذج التيكسوتروب. هذه الطريقة تسمح لنا بتقسيم فضاء المسألة الإبتدائية إلى عدة فضاءات جزئية صغيرة مرفقة بمسائل جزئية حلها يكون بطريقة سريعة ، حيث أن حل المسألة الأصلية نحصل عليه من خلال حلول المسائل الجزئية المرافقة لها.

إن الصعوبات الأساسية في حل نموذج التيكسوتروب بطريقة تفكيك النطاق تكمن في السلوك غير المنتظم لحواف النطاقات الجزئية مع الحفاظ على عدم إلاستمرارية. لتجاوز هذه الصعوبات نقوم بتسوية المسألة الأصلية ثم نطبق طريقة تفكيك النطاقات بتغطية أو بدون تغطية، لإثبات فعالية الخوارزمية نقوم بتطبيقه على تحسين الصور و مقارنة النتائج المحصل عليها مع النتائج المحصل عليها بنفس الخوارزمية لحل مسألة التباين الكلي.

INTRODUCTION

Image restoration is one of the fundamental and challenging tasks in image processing [3], and phenomenal advances have been achieved in variational and partial differential equations (PDE)-based approaches since the seminal work [59]. The Rudin Osher and Fatemi (ROF) model minimizes the total variation (TV) over the space of bounded variation (BV), so it is capable of preserving sharp edges and boundaries with a high quality recovery. More precisely, given a bounded image domain $\Omega \subseteq \mathbb{R}^d$, $(d = 1, 2, 3)$, one is interested in the general minimization problem:

$$\min_{u \in BV} \left\{ \alpha \int_{\Omega} |\nabla u| + \int_{\Omega} f(u) dx \right\}, \quad \alpha > 0 \tag{1}$$

where the gradient is presented in the distributional sense [39], and $f(.)$ as a differentiable functional. The associated Euler-Lagrange equation takes the form

$$-\alpha \operatorname{div} \left(\frac{\nabla u}{|\nabla u|} \right) + f'(u) = 0, \tag{2}$$

which is also known as the curvature equation. As the TV model (1-2) continues to enjoy applications in diverse areas such as image denoising, debluring and segmentation, [74], interface evolution [56], and inverse problems [24], there still exists a great demand for developing fast and robust methods for such minimization problems and nonlinear PDEs, although considerable progress has been made in several directions. Among others, existing methods in the literatures can be classified into the following types:

1. The gradient descent method [19] , [25]: Instead of solving the nonlinear PDE, it involves (2) with an artificial time and minimizes the energy along the gradient descent direct via the evolution of a parabolic equation. This approach is very reliable, but converges considerably slowly.
2. The method using diffusivity fixed-point iteration (see, e.g.,[67], [68]): It solves the linearized version of the nonlinear steady-state PDE (2) iteratively by treating the nonlinear term $1/|\nabla u|$ explicitly. Various iterative solvers have been considered, but further studies are still needed, in particular, techniques to speed up the outer solvers for large size images.

3. The dual approach (cf. [12], [23]): it introduces a dual variable (the original unknown function u in(1) is referred to as the primal variable). These methods overcome the non-differentiability of the cost functional in (1). They often lead to more efficient algorithms, and have received increasing interests recently.

4. Additive operator splitting (AOS) scheme: this type of schemes was first developed for (nonlinear elliptic/parabolic) monotone equation and Navier-Stokes equations in [33]. In image processing applications, the AOS scheme was found to be an efficient way for approximating the PeronaMalik filter [72], [73], especially if symmetry in scale-space is required. The AOS scheme is first order in time, semi-implicit, and unconditionally stable with respect to its time-step [73]. These methods have been applied to a wide range of image processing applications and often lead to very efficient numerical algorithms.

5. Bregman iteration: Iterative optimization methods based on penalization or Bregman distance [69], [71] have been proposed very recently. In [71], [69], the authors used variable-splitting to separate theL^1 and L^2 terms and then solved an equality constrained optimization problem by penalization and alternative minimization. Bregman iteration for image processing was originated from [12] and was introduced by Osher et. al. in [19]. It has been extended to wavelet-based denoising [70], nonlinear inverse scale space in [24] and compressed sensing [76]. The basic ideas to transform the equality constrained optimization problem into a series of unconstrained problems using Bregman distance. By combining the variable-splitting and Bregman iteration, Goldsteinet. al. obtained split Bregman method in [12] which is particularly efficient for L^1 regularized problems, e.g., TV restoration.

6. Augmented Lagrangian method [76]: for total variation image restoration. It has many advantages over other methods such as penalty method. As only linear problems need to be solved during the iterations, FFT can be applied to get extremely efficient implementations. In addition, the augmented Lagrangian approach provides close connections to dual methods and split Bregman iteration [76].

7. Multigrid method [75]: It is one of the most powerful numerical methods for solving some linear and nonlinear partial differential equations. In [58], the linear algebraic multigrid method [59] was adopted for solving the above PDE in each (outer) step of

a fixed iteration,while attempted to use the standard multigrid methods with a non-standard and some what global smoother. Recently, nonlinear multigrid methods based on the subspace correction approach, for example in [58] have been introduced to image processing in [24]. Numerical experiments indicate their promising numerical potentials.

These methods have been widely used for image processing, and their strength and weakness have also been observed from real applications. Dual methods and Bregman iterations are fast, but they are under intensive investigation for the applications to more general image processing problems. Graph-cut approaches are usually fast, but they can be only applied to a special class of problems and could also have matriculation errors. The AOS and multigrid methods also have limitations in the models that they can be applied.

The purpose of this work is to apply a fast solver based on overlapping domain decomposition and a coarse mesh correction for image processing tasks. Our aim is to demonstrate several essential advantages of the implemented method. More precisely,

1. This method can be used for various general variational-based image processing problems. Indeed, based on this notion, one can easily apply the existing solvers to the minimization problem by solving a sequence of subdomain problems of smaller scale.
2. In practice, the original problem, e.g., large size 3D data processing, could be too large, which induces difficulties in applying a given solver. By splitting a large problem into many smaller sub-problems, that we can easily solve.
3. The implemented method can save CPU time cost. The gain is significant, efficient and relatively accurate for subdomain solvers,
4. The implemented method is well-suited for distributed-memory in parallel computers.

It is known that domain decomposition (DD) methods are powerful iterative methods for solving partial differential equations [75]. Some recent progress has shown that DD methods are also efficient for some nonlinear elliptic problems and some convex minimization problems with mesh independent convergence. So far, it is still unknown wether or not, we can use domain decomposition methods for the ROF model. Some recent efforts have been devoted to study these problems [56]. For simplicity of presentation, we implemented and tested this method on the tixotrop model, see (1), and presented some results. We provide numerical

results to show its capability in processing images of large size with low cost in CPU time and memory. Once again, the essence of this technique is to view a domain decomposition procedure in space. The original minimization problem related to the tixotrop model is reduced to some sub-minimization problems with smaller size over the sub-domains. If the sub-minimization problems are solved exactly, the convergence of the generated sequence is trivial to prove. Due to the degeneracy of the nonlinear equation of tixotrop, it is not convincing that we will be able to prove the convergence rate for the numerical solutions.

This thesis is organized as follows. In chapter one, we present the preliminary results used in our study. In Chapter 2 we write down the method in a more general setting and start with the description of the subspace correction algorithm for the TV technique.

Contribution of this thesis

The main contribution of this thesis reflected in [77] is presented in chapter 3. We use in this work a fast algorithm for nonlinear minimization problems with particular applications to image denoising. We describe in details the implementation of the domain decomposition and coarse mesh correction techniques. We compare the non-overlapping domain decomposition approach with the total variation minimization, according more attention to the computational time and the efficiency.

1.0 PRELIMINARIES

In his chapter, we present a basic notion and principal mathematical results which will be useful to us in the next chapters. Further details can be found in [3], and [47]

1.1 DEFINITIONS AND BASIC PROPERTIES

Definition 1. *An open connected set* $\Omega \subseteq \mathbb{R}^n$ *is called a domain. By* $\bar{\Omega}$ *we denote the closure of* Ω*;* $\partial\ \Omega = \Gamma$ *is the boundary.*

We use the following notation:

$$x = (x_1, x_2, ..., x_n) \in \mathbb{R}^n, \quad \partial_j u = \frac{\partial u}{\partial x_j},$$

$$\alpha = (\alpha_1, \alpha_2, ..., \alpha_n) \in \mathbb{Z}_+^n, \quad \textit{is a multi-index}$$

$$|\alpha| = \alpha_1 + \alpha_2 + ... + \alpha_n, \ \partial^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} ... \partial x_n^{\alpha_n}}$$

$$\textit{Next, } \nabla u = (\partial_1 u, ..., \partial_n u), \quad |\nabla u| = \left(\sum_{j=1}^{n} |\partial_j u|^2 \right)^{\frac{1}{2}}$$

$$\textit{and} \qquad Du = (D_1 u, ..., D_N u) \quad \textit{in} \ \ \Omega$$

Definition 2. *Let* $1 \leq p \leq +\infty$*, by* $L^p(\Omega)$ *we denote the Banach space of all measurable functions on* Ω*, such that:*

$$L^p(\Omega) = \left\{ u : \Omega \longrightarrow \mathbb{R}; u \ \ \textit{measurable and} \int_\Omega |u|^p \, dx < +\infty \right\}$$

with the associated norm

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega |u|^p \, dx \right)^{\frac{1}{p}}$$

For $p = \infty$, $L^\infty(\Omega)$ denotes the essentially bounded functions

$$\|u\|_{L^\infty(\Omega)} = \sup_{x \in \Omega} |u(x)| \, .$$

For $p = 2$, the space $L^2(\Omega)$ is a Hilbert space, with inner-product

$$\forall u, v \in L^2(\Omega), \ \langle u, v \rangle_{L^2(\Omega)} = \int_\Omega u(x) v(x) dx$$

1.1.1 Weak derivatives

Let $C_0^\infty(\Omega)$ denotes the space of infinitely differentiable functions $\varphi : \Omega \longrightarrow \mathbb{R}$, with compact support in Ω. We will call a function φ belonging to $C_0^\infty(\Omega)$, a test function.

Definition 3. *Suppose that $\alpha \in \mathbb{N}^n$ is a multi-index. A function $f \in L^1_{loc}(\Omega)$ has a weak derivative $\partial^\alpha f \in L^1_{loc}(\Omega)$ if*

$$\int_\Omega \partial^\alpha f(x) \varphi(x) dx = (-1)^{|\alpha|} \int_\Omega f(x) \partial^\alpha \varphi(x) dx, \quad \textit{for all } \varphi(x) \in C_0^\infty(\Omega).$$

1.1.2 Sobolev spaces

Definition 4. *We call a Sobolev space of order 1 on Ω, the space*

$$H^1(\Omega) = \left\{ u \in L^2(\Omega), \partial_i u \in L^2(\Omega), \ 1 \leq i \leq n \right\}$$

The space $H^1(\Omega)$ is endowed with the norm associated to the inner product:

$$\langle u, v \rangle = \int_\Omega \left(uv + \sum_{i=1}^n \partial_i u \partial_i v \right) dx,$$

and we note the corresponding norm:

$$\|v\| = \sqrt{\langle u, v \rangle} = \left(\int_\Omega |u|^2 dx + \int_\Omega |\nabla u|^2 \right)^{1/2}.$$

These spaces are separable Hilbert spaces.

Definition 5. *Let $1 \leq p < \infty$ and for every $m \in \mathbb{N}$, $m \geq 1$, the Sobolev space $W^{m,p}(\Omega)$ is defined to be*

$$W^{m,p}(\Omega) = \left\{u \in L^p(\Omega), \partial^\alpha u \in L^p(\Omega),\ \forall \alpha \in \mathbb{N}^d,\ |\alpha| \leq m\right\},$$
where $\partial^\alpha u$ is the derivative in the distribution sense

endowed with the norm:

$$\|u\|_{W^{m,p}} = \left(\sum_{|\alpha| \leq m} \int_\Omega |\partial^\alpha u|^p \, dx\right)^{\frac{1}{p}}.$$

All these spaces are Banach spaces. However, we will consider here the spaces $W^{m,2}(\Omega) = H^m(\Omega)$.

Definition 6. *We denote by $H_0^1(\Omega)$ the closure of $D(\Omega)$ in $H^1(\Omega)$. By extension, we note $H_0^m(\Omega)$ the closure of $D(\Omega)$ in $H^m(\Omega)$ (for the norm $\|.\|_{H^m(\mathbb{R}^d)}$).*

1.1.3 Convex analysis

Let E be a norm vector space over $\mathbb{R}$ (n.v.s).

Definition 7. *The set,*

$$dom(f) = \{x \in E/ \quad f(x) < +\infty\}$$

is the effective domain of f.

Definition 8. *If f does not assume both $-\infty$ and $+\infty$ as values this definition of a convex function is equivalent to*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2),\ \forall x_1, x_2 \in E, \quad and \quad 0 \leq \lambda \leq 1.$$

Definition 9. *A convex function f on E is lower semi-continuous (1.s.c.) if, for each $\mu \in \mathbb{R}$, the convex level set*

$$\{x \in E/ \quad f(x) \leq \mu\}$$

is a closed set in E. Lower semi-continuity of convex functions is a constructive property.

Given any convex function f on E, we may construct a *1.s.c.* convex function f on E by taking f such that

$$f(x) = \lim_{z \longrightarrow} \inf f(z), \forall x \in E$$

.

Definition 10. *A convex function f on E is said to be proper if $f(x) > -\infty$ for all $x \in E$ and $f(x) < +\infty$ for at least one $x \in E$.*

1.1.3.1 Conjugate Convex Functions

Definition 11. *Let f be a proper convex function on E. Its conjugate function f^* on E^* (with respect to the givern-bilinear function $\langle ., . \rangle$) is defined by*

$$f^*(x^*) = \sup_{x \in E} \{(x, x^*) - f(x)\}, \forall x \in E^*$$

with E^* is called the topological dual of E that is, the space of all continuous linear functionals on E; the (dual) norm on E^* is defined by

$$\|f^*\|_{E^*} = \sup_{\substack{x \in E \\ \|x\| < 1}} |f(x)|$$

The function f^* is a *1.s.c.* convex function but not necessarily proper. However, if f is a *l.s.c*, proper convex function, then f^* is also *l.s.c*, proper convex and $(f^*)^* = f$

1.2 SUBDIFFERENTIABILITY

In this section we make use results of [37].

Definition 12. *Let $f : E \to \bar{\mathbb{R}}$ be a real-valued functional on a Banach space E, The directional derivative of f at $x \in E$ in the direction $y \in E$ is defined as the limit, if there exists,*

$$f'(x, y) = \lim_{\lambda \to 0^+} \frac{f(x + ty) - f(x)}{t}$$

the last expression is called the Gâteau differential of $f(x)$ at $x \in E$, and is denoted by $f'(x) \in E^*$, if the above limit exists for every $y \in E$ and

$$f'(x,y) = \langle y, f'(x) \rangle .$$

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 13. *Let $f : E \to \mathbb{R}$ be a convex function, E^* its topological dual, $\langle \cdot, \cdot \rangle$ the bilinear canonical pairing over $E \times E^*$. The subdifferential of f at $x \in E$, is defined by*

$$\partial f(x) = \begin{cases} \emptyset & \text{if} \quad f(x) = \infty, \\ \{x^* : \langle x^*, y - x \rangle + f(x) \leq f(y) \quad \text{for all } y \in E\} & \text{otherwise} \end{cases}$$

$y \in \partial f(x)$ is called a subgradient. It is obvious from this definition that $0 \in \partial f(x)$ if and only if x is a minimizer of f.

Proposition 14. *For every function $f : E \to \bar{\mathbb{R}}$ we have*

$$x^* \in \partial f(x) \implies x \in \partial f^*(x^*) .$$

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

$$x^* \in \partial f(x) \iff x \in \partial f^*(x^*) . \tag{1.1}$$

1.2.1 Subdifferential Calculus

Let E be a locally convex space, $f : E \to \bar{\mathbb{R}}$, and $t > 0$. At every point $x \in E$, we have

$$\partial(tf)(x) = t\partial f(x).$$

Moreover, let $f_1, f_2 : E \to \mathbb{R}$. At every point $x \in E$, we have

$$\partial f_1(u) + \partial f_2(u) \subset \partial(f_1 + f_2)(u)$$

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

Proposition 15. *Let f_1 and f_2 be convex, l.s.c., and proper. If there exists a point $\bar{x} \in Dom f_1 \cap Dom f_2$ where f_1 is continuous, then we have for all $u \in E$*

$$\partial(f_1 + f_2)(x) = \partial f_1(x) + \partial f_2(x).$$

Let us consider now the subdifferential of a composite function.

Proposition 16. *Let E_1 and E_2 be two locally convex sets with topological duals E_1^* and E_2^*, $g : E_1 \to E_2$ with adjoint g^*, $f : E_2 \to \mathbb{R}$ a convex, l.s.c., and proper function, and $g \circ f : E_1 \to \mathbb{R}$ also a convex, l.s.c., and proper function. If there is a point $g(\bar{x})$, for $\bar{x} \in E_1$, where f is continuous and finite, then for all points $x \in E_1$, we have*

$$\partial (f \circ g)(x) = g^* \partial f (g(x)).$$

1.3 FUNCTIONS OF BOUNDED VARIATION

In image processing, one is interested in recovering and preserving discontinuities in images. Using classical Sobolev spaces, denoted by $W^{1,1}$ (the Sobolev space of L^1-functions with L^1-distributional derivatives), does not allow us 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 17. *Let Ω be an open subset of $\mathbb{R}^N$. A function $u \in L^1(\Omega)$ whose partial derivatives in the sense of distributions are measures with finite total variation in Ω is called a function of bounded variation. The vector space of functions of bounded variation in Ω is denoted by $BV(\Omega)$. Thus $u \in BV(\Omega)$ if and only if $u \in L^1(\Omega)$ and there are Radon measures $1, ..., N$ with finite total mass in such that*

$$\int_\Omega u \frac{\partial \phi}{\partial x_i} dx = -\int_\Omega \phi d\ D_i u \quad \text{for all } \phi \in C_c^\infty(\Omega), \quad i = 1, ..., N \tag{1.2}$$

If $u \in BV(\Omega)$, the total variation of the measure Du is

$$\|Du\| = \sup \left\{ \int_\Omega u \operatorname{div} \phi dx,\ \phi \in C_c^1\left(\Omega, \mathbb{R}^N\right), \quad |\phi(x)| \leq 1 \quad \text{for,} \quad x \in \Omega \right\} < \infty$$

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

$$\|u\|_{BV} = \int_\Omega |u|\, dx + \|Du\|$$

is a Banach space. We also use $\int_\Omega |u|\, dx$ to denote the total variation $\|Du\|$.

Proposition 18. *Let $u \in L^1_{loc}(\Omega)$. Then, u belongs to $BV(\Omega)$ if and only if $\|Du\| < \infty$. In addition, $\|Du\|$ coincides with $|Du|$ for any $u \in BV(\Omega)$ and $u \longmapsto |Du|(\Omega)$ is l.s.c. in $BV(\Omega)$ with respect to the $L^1_{loc}(\Omega)$ topology.*

Proposition 19. *1. (lower semi-continuity of the total variation) Suppose $u_n \in BV(\Omega)$, $n = 1, 2, ...$ and that $u_n \to u$ in $L^1_{loc}(\Omega)$. Then*

$$\int_\Omega |Du| \leq \lim_{n \to +\infty} \inf \int_\Omega |Du_n|$$

2. (approximation by smooth functions) Assume that $u \in BV(\Omega)$. There is a sequence of functions $u_n \in BV(\Omega) \cap C^\infty(\Omega)$ such that

a. $u_n \to u$ in $L^1(\Omega)$ and

b. $\int_\Omega |Du_n| \to \int_\Omega |Du| \quad$ as $n \to +\infty$

Moreover, if $u \in BV(\Omega) \cap L^p(\Omega)$, $p < \infty$, we can find $u_n \in L^p(\Omega)$, $u_n \to u$ in $L^p(\Omega)$.

Definition 20. *Let $u_n, u \subset BV(\Omega)$. Then $(u_n)_n$ converges weakly-$*$ to u in $BV(\Omega)$ if $u_n \to u$ in $L^1_{loc}(\Omega)$ and Du_n converges weakly-$*$ to Du as measures in Ω.*

Theorem 21. *1. Let $u_n, u \in BV(\Omega)$. Then $u_n \to u$ weakly-$*$ in $BV(\Omega)$ if and only if u_n is bounded in $BV(\Omega)$ and converges to u in $L^1_{loc}(\Omega)$.*

2. (compactness) Let $\Omega \subset \mathbb{R}^N$ be open, bounded, with $\partial\Omega$ Lipschitz. Assume $u_n \in BV(\Omega)$ satisfying $\|u_n\|_{BV(\Omega)} < M < \infty$ for all $n \geq 1$. Then there is a subsequence u_{n_j} and a function $u \in BV(\Omega)$ such that $u_{n_j} \to u$ in $L^1(\Omega)$

1.4 TOTAL VARIATION MINIMIZATION

Total variation based image restoration models were first introduced by ROF in their pioneering work [56] on edge preserving image denoising. It was designed with the explicit goal of preserving sharp discontinuities (edges) in images while removing noise and other unwanted fine scale detail.

Let $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ an original image descrbing a real scene and let u_0 be the observed image of the same scene (i.e. a degradation of u). We assume that:

$$u_0 = Ru + \varepsilon \tag{1.3}$$

where ε stands for a white additive some noise and R is a linear operator modeling the image-formation device. Given u_0, the problem is then to reconstruct u knowing 1.3. As we will see, the problem is ill-posed and we are only able to carry out an approximation of u.

A classical way to overcome ill-posed minimization problems is to add a regularization term to the energy. This idea was introduced in 1977 by Tikhonov et al *[60]*. The authors proposed to consider the minimization problem:

$$F(u) = \int_\Omega |u_0 - Ru|^2 \, dx + \lambda \int_\Omega |\nabla u|^2 \, dx. \tag{1.4}$$

The first term in $F(u)$ measures the fidelity to the data. The second one is a smoothing term and $\lambda > 0$ is a fixed regularization parameter weighting the importance of the two terms.

To study this problem, the fonctional space for which both terms are well-defined is $W^{1,2}(\Omega) = \left\{u \in L^2(\Omega) : \nabla u \in [L^2(\Omega)]^2\right\}$,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 1.4 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.

In the context of image restoration Rudin, Osher and Fatemi [56] proposed to use the total variation as a regularization technique. We recall that for $u \in L^1_{loc}(\Omega)$

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

is the variation of u, where $C^1_c(\Omega)$ denotes the space of C-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], if and only if $TV(u) < \infty$. In this case, $|Du|(\Omega) = TV(u)$, where $|Du|$ 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. Additionally it is convex and therefore also the minimization problem, which reads then as

$$\arg\min_u \int_\Omega |Ru - u_0|^2 dx + 2\alpha |Du|(\Omega),$$

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

$$\Im(u) := \|Ru - u_0\|^2_{L^2(\Omega)} + 2\alpha |Du|(\Omega), \tag{1.5}$$

where $R : L^2(\Omega) \to L^2(\Omega)$ is a bounded linear operator, $u_0 \in L^2(\Omega)$ is a datum, and $\alpha > 0$ is a fixed regularization parameter. More precisely, we are concerned with minimizing J by means of subspace correction and domain decomposition. That means, instead of minimizing 1.5 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 [23]. Several numerical strategies to efficiently perform total variation minimizations have been proposed in the literature *see for example [47].*

2.0 METHOD OF DOMAIN DECOMPOSITION

2.1 BASIC IDEA OF SUBSPACE CORRECTION AND DOMAIN DECOMPOSITION

The method of subspace corrections are general iterative methods that have a variety of applications. The method of alternating projection, first proposed by Von Neumann (1933) (see [62]), is an algorithm for finding the best approximation to any given point in a Hilbert space from the intersection of a finite number of subspaces. The method of subspace corrections, an abstraction of general linear iterative methods such as multigridand domain decomposition methods, is an algorithm for finding the solution of a linear system of equations. By contrast for non-smooth and non-additive energies, such as 1.5, 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 the thesis of [47] the author showed that decomposition strategies may converge to a minimizer of the original problem for non-smooth and non-additive cases. Theorem 2.2.8 and Theorem 2.3.1, and 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 [64] for more details on subspace correction methods.

The first known subspace correction strategy was proposed by H. A. Schwarz (1869)(see [57]) 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 [[55], $p26$]. 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 [[25], [55], [59]] for more details. For simplicity we discuss these methods now for a simple problem "the Poisson problem", i.e., second-order self-adjoint elliptic problem,

$$\mathcal{L}u = -\Delta u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on } \partial\Omega. \tag{2.1}$$

Let Ω be a bonded domain in $\mathbb{R}^n$ ($n = 2$ or 3) ,with a lipschitz boundary Γ.

2.1.1 Non-overlapping Domain Decomposition

Let us start by splitting the spatial domain Ω into two non-overlapping subdomains $\Omega_1$and Ω_2 with interface Γsuch that $\overline{\Omega} = \overline{\Omega_1 \cup \Omega_2}$ and $\Omega_1 \cap \Omega_2 \neq \emptyset$, (see Figure 1.5.). We define the domaine Ω to be:

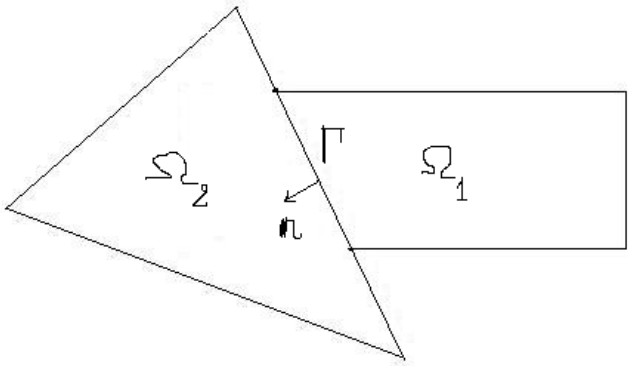

Figure 1.1. Non-overlapping decomposition into two domains

The interface between these two regions is defined by $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$. In addition, we assume that the boundaries of the subdomains are regular enough. Then problem (2.1) can be formulated as

$$\begin{cases} \mathcal{L}u_1 = f \ \text{ in } \Omega_1 \\ u_1 = 0 \ \text{ on } \partial\Omega_1 \cap \partial\Omega \\ \begin{cases} u_1 = u_2 \ \text{ on } \Gamma \\ \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} \ \text{ on } \Gamma \end{cases} \\ \mathcal{L}u_2 = f \ \text{ in } \Omega_2 \\ u_2 = 0 \ \text{ on } \partial\Omega_2 \cap \partial\Omega, \end{cases} \tag{2.2}$$

where each n is the outward pointed normal on Γ from Ω_1. Here we see that due to the partition of the original problem (2.1) 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 (2.1) and the multi-domain problem (2.2) is in general not obvious, but can be shown under suitable regularity assumptions on f, typically $f \in L^2$, by considering the associated variational formulation.

Iterative Methods

We will now focus on solving the multi-domain problem (2.2) 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 Method by Agoshkov and Lebedev The following non-overlapping domain decomposition algorithm was proposed by Agoshkov and Lebedev (see [1]): given $u_1^{(0)}$ and $u_2^{(0)}$, for each $k \geqslant 0$ we have to solve

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

and

$$\begin{aligned} & \left\{ \begin{array}{cc} \mathcal{L}u_2^{(K+1)} = f & \text{in } \Omega_2, \\ u_2^{(K+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma, \\ \frac{-q_k \partial u_2^{(K+1/2)}}{\partial n} + u_2^{(K+1/2)} = -q_k \frac{\partial u_1^{(k+1)}}{\partial n} + u_1^{(k+1)} & \text{on } \Gamma, \end{array} \right. \\ u_2^{(k+1)} &= u_2^{(k)} + \beta_{k+1}\left(u_2^{(K+1/2)} - u_1^{(k)}\right) \quad \text{in } \Omega_2, \end{aligned} \tag{2.3}$$

where $p_k, q_k \geqslant 0$ and α_{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 (see [47]), which is obtained by setting $p_k = \gamma_1$, $q_k = 1/\gamma_2$ and $\alpha_k = \beta_k = 1$ in (2.3).

2.1.2 Overlapping Domain Decomposition

In this section we describe the so-called multiplicative and additive Schwarz methods (see [57])

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.

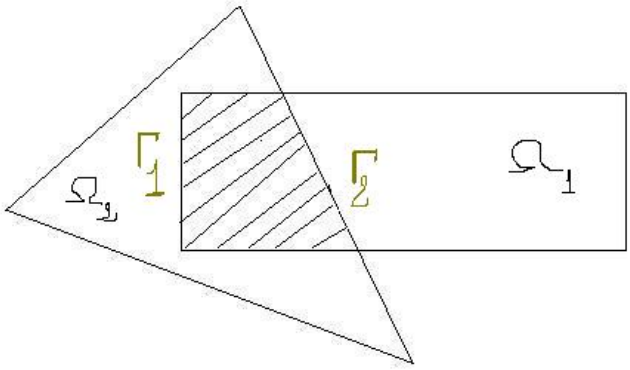

Figure 1.2. Overlapping decomposition into two domains

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 $u^{(1)}, u^{(2)}, ...$ by solving

$$\begin{cases} \mathcal{L}u_1^{(k+1)} = f \quad \text{in} \quad \Omega_1, \\ u_1^{(k+1)} = u_{1/\Gamma_1} \text{ on } \Gamma_1, \\ u_1^{(k+1)} = 0 \text{ on } \partial\Omega_1 \setminus \Gamma_1. \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{L}u_2^{(k+1)} = f \quad \text{in} \quad \Omega_2, \\ u_2^{(k+1)} = u_{1/\Gamma_2} \text{ on } \Gamma_2, \\ u_2^{(k+1)} = 0 \text{ on } \partial\Omega_2 \setminus \Gamma_2. \end{cases} \tag{2.4}$$

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 (2.4) converges to a solution of problem (2.1), see [49] and for a variational based proof consult [55] 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 $k \geqslant 0$

$$\begin{aligned} \left\| u_{/\Omega_1} - u_1^{(k+1)} \right\|_{L^\infty(\Omega_1)} &\leq c_1^k c_2^k \left\| u - u^{(0)} \right\|_{L^\infty(\Gamma_1)}, \\ \left\| u_{/\Omega_2} - u_2^{(k+1)} \right\|_{L^\infty(\Omega_2)} &\leq c_1^{k+1} c_2^k \left\| u - u^{(0)} \right\|_{L^\infty(\Gamma_2)}. \end{aligned}$$

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 [47].

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

$$J(w,u) := \frac{1}{2} a(w,w) - (f,w) + a(u,w). \tag{2.5}$$

Let us rewrite (2.4) in the following form

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

and

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

The variational formulation of method (2.4) reads as follows: initialize $u^{(0)} \in H_0^1(\Omega)$, and for $k \geqslant 0$ solve

$$\begin{aligned} w^{(k)}1 &\in H_0^1(\Omega_1) : a(w_1^{(k)}, v_1) = (f, v_1) - a(u^{(k)}, v_1) \text{ for all } v_1 \in H_0^1(\Omega_1), \\ u^{(k+1/2)} &= u^{(k)} + w_1^{(k)}, \\ w_2^{(k)} &\in H_0^1(\Omega_2) : a(w_2^{(k)}, v_2) = (f, v_2) - a(u^{(k+1/2)}, v_2) \text{ for all } v_2 \in H_0^1(\Omega_2), \\ u^{(k+1)} &= u^{(k+1/2)} + w_2^{(k)}. \end{aligned} \tag{2.6}$$

Or equivalently

$$\left\{\begin{array}{c} w_1^{(k)} = \arg\min\limits_{w_1 \in H_0^1(\Omega_1)} J(w_1, u^{(k)}), \\ u^{(k+1/2)} = u^{(k)} + w_1^{(k)}, \\ w_2^{(k)} = \arg\min\limits_{w_2 \in H_0^1(\Omega_2)} J(w_2, u^{(k+1/2)}), \\ u^{(k+1)} = u^{(k+1/2)} + w_2^{(k)}. \end{array}\right. \tag{2.7}$$

From (2.6) we have

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

Which means

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

Or equivalently

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

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

$$u - u^{(k+1)} = (I - P_2)(u - u^{(k+1/2)}) = (I - P_2)(I - P_1)(u - u^{(k)}) \text{ for all } k \geqslant 0. \tag{2.8}$$

Additive Schwarz Method

If we make the two steps (2.4) independent from each other, then we obtain the additive alternating Schwarz method, which computes the sequence of approximations by solving

$$\left\{\begin{array}{c} \mathcal{L}u_1^{(k+1)} = f \ \text{ in } \Omega_1 \\ u_1^{(k+1)} = u_{|\Gamma 1}^{(k)} \ \text{ on } \Gamma_1 \\ u_1^{(k+1)} = 0 \text{ on } \partial\Omega_1 \backslash \Gamma_1 \end{array}\right. \quad \text{and} \quad \left\{\begin{array}{c} \mathcal{L}u_2^{(k+1)} = f \ \text{ in } \Omega_2 \\ u_2^{(k+1)} = u_{|\Gamma 2}^{(k)} \ \text{ on } \Gamma_2 \\ u(k+1)2 = 0 \ \text{ on } \partial\Omega_2 \backslash \Gamma_2. \end{array}\right. \tag{2.9}$$

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

$$u^{(k+1)}(x) = \begin{cases} u_1^{(k+1)}(x) & x \in \Omega \backslash \Omega_2, \\ u_1^{(k+1)}(x) + u_2^{(k+1)}(x) - u^{(k)}(x) & x \in \Omega_1 \cap \Omega_2, \\ u_2^{(k+1)}(x) & x \in \Omega \backslash \Omega_1. \end{cases} \tag{2.10}$$

Variational Formulation The variational formulation of method (2.9) reads as

$$\begin{cases} w_1^{(k)} \in H_0^1(\Omega_1) : a(w_1^{(k)}, v_1) = (f, v_1) - a(u^{(k)}, v_1) \text{ for all } v_1 \in H_0^1(\Omega_1), \\ w_2^{(k)} \in H_0^1(\Omega_2) : a(w_2^{(k)}, v_2) = (f, v_2) - a(u^{(k)}, v_2) \text{ for all } v_2 \in H_0^1(\Omega_2), \\ u^{(k+1)} = u^{(k)} + w_1^{(k)} + w^{(k)}. \end{cases} \tag{2.11}$$

Or

$$\begin{cases} w_1^{(k)} = \arg\min_{w_1 \in H_0^1(\Omega_1)} J(w_1, u^{(k)}), \\ w_2^{(k)} = \arg\min_{w_2 \in H_0^1(\Omega_2)} J(w_2, u^{(k)}), \\ u^{(k+1)} = u^{(k)} + w_1^{(k)} + w_2^{(k)}. \end{cases} \tag{2.12}$$

Where $J(w, u) = \frac{1}{2}a(w, w) - (f, w) + a(u, w)$. By relation (2.10) we verify that the original formulation (2.9) is equivalent to the variational formulation. Moreover from (2.11) 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

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

Or equivalently

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

Then by using the update (2.10) we get the following error recursion formula:

$$u - u^{(k+1)} = u - u_1^{(k+1)} - u_2^{(k+1)} - u^{(k)} = (I - P_1 - P_2)(u - u^{(k)}) \quad \text{for all} \quad k \geqslant 0. \tag{2.13}$$

Inspired by the variational formulation (2.7) and (2.12) of the multiplicative and additive Schwarz method in [42], 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.

2.1.3 Subspace Correction for ℓ_1-norm Minimization

The minimization of the ℓ_1-norm is well-known to give an effective way for reconstructing sparse signals from linear measurement [42],[47]. It has been shown that the minimization of the ℓ_1-norm is very effective in several applications, such as compressed sensing, and image processing [21], [17], [18]...

Let H be a real separable Hilbert space and for a countable index set A we define $\ell_p(A) := \{u = (u_\lambda)_{\lambda\in A} : (\sum_{\lambda\in A} |u_\lambda|^p)^{1/p}\}$ for $1 \leqslant p < \infty$. We are interested in the numerical minimization in $\ell_2(A)$ of the functional

$$J(u) := \|Tu - g\|_H^2 + 2\alpha \|u\|_{\ell_1(A)}, \tag{2.14}$$

where $T : \ell_2(A) \to H$ is a bounded linear operator, $g \in H$ 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 [31]: pick an initial $u^{(0)} \in \ell_2(A)(u^{(0)} = 0$ is a good choice) and iterate

$$u^{(n+1)} = S_\alpha(u^{(n)} + T^*(g - Tu^{(n)})), n \geqslant 0, \tag{2.15}$$

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

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

is the so-called soft-thresholding operator. The strong convergence of the algorithm in (2.15) to find minimizers of J is proved in [31]. In [13], it was shown that under additional conditions on the operator T or on minimizers of (2.14) the algorithm in (2.15) converges linearly, although with a rather poor rate in general, see [39] for a more detailed discussion. There exist several alternative approaches, that promise to solve ℓ_1-minimization with a fast convergence [8]. One way to accelerate the speed of convergence of minimizing iterative soft-thresholding algorithms for large-scale problems was proposed in [39], where 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.

2.1.4 Sequential Algorithm

We decompose the domain A: set A into two disjoint sets A_i $(i = 1, 2)$ i.e.$A = A_1 \cup A_2$. Associated with this decomposition we define $Vi = \{u_A \in \ell_2(A) : supp(u_A) \subset A_i\}$ for $i = 1, 2$. Then we minimize J in (2.14) by using the following alternating algorithm: take an initial $u^{(0)} = u^{(0)}_{A_1} + u^{(0)}_{A_2} \in V_1 \oplus V_2$, for example, $u^{(0)} = 0$, and iterate

$$\begin{aligned} u^{(n+1)}_{A_1} &\simeq \arg\min_{u_{A_1} \in V_1} J(u_{A_1} + u^{(n)}_{A_2}), \\ u^{(n+1)}_{A_2} &\simeq \arg\min_{u_{A_2} \in V_2} J(u^{(n+1)}_{A1} + u_{A_2}), \\ u^{(n+1)} &:= u^{(n+1)}_{A_1} + u^{(n+1)}_{A_2}, \end{aligned} \tag{2.16}$$

where u_{A_i} is supported on A_i only, $i = 1, 2$. This algorithm is inspired by (2.7) and (2.12), but differently from the situations there, the energy (2.14) is now nonsmooth. Nevertheless we observe that the ℓ_1-norm splits additively

$$\|u_{A_1} + u_{A_2}\|_{\ell_1(A)} = \|u_{A_1}\|_{\ell_1(A_1)} + \|u_{A_2}\|_{\ell_1(A_2)},$$

and hence the subproblems in (2.16) are of the same kind as the original problem (2.14), i.e., for example, for the problem on A_1 we have

$$\arg\min_{uA_1 \in V_1} J(u_1 + u^{(n)}_2) = \arg\min_{uA_1 \in V_1} \left\| T_{A_1} u_{A_1} - (g - T_{A_2} u^{(n)}_{A_2}) \right\|_{L^2(\Omega)} + 2\alpha \|u_{A_1}\|_{\ell_1(A_1)},$$

where T_{A_i} $(i = 1, 2)$ are the restrictions of the matrix T to the columns indexed by A_i. Moreover, this splitting results in a dimension reduction for each subproblem. For solving the subminimization problems of (2.16) we can use one of the before mentioned methods, for example, again the iterative thresholding algorithm:

$$u^{(n+1,\ell+1)}i = S_\alpha(u_{A_1}^{(n+1,\ell)} + T_{Ai}^*((g - T_{A_i}u_{A_i}^{(n)}) - T_{A_i}u^{(n+1,\ell)})), \widehat{i} \in \{1, 2\}\backslash\{i\}. \tag{2.17}$$

This leads to the following sequential algorithm: pick an initial $u(0) = u_{A_1}^{(0,L)} + u_{A_2}^{(0,M)} \in V_1 \oplus V_2$, for example, $u^{(0)} = 0$, and iterate

$$\begin{cases} \begin{cases} u_{A_1}^{(n+1,0)} = u_{A_1}^{(n,L)}, \\ u_{A_1}^{(n+1,\ell+1)} = S_\alpha\left(u_{A_1}^{(n+1,\ell)} + T_{A_1}^*((g - T_{A_2}u_{A_2}^{(n,M)}) - T_{A_1}u_{A_1}^{(n+1,\ell)})\right), \\ \ell = 0, ..., L-1, \end{cases} \\ \begin{cases} u_{A_2}^{(n+1,0)} = u^{(n,M)}, \\ u_{A_2}^{(n+1,\ell+1)} = S_\alpha\left(u_{A_2}^{(n+1,\ell)} + T_{A_2}^*((g - T_{A_1}u_{A_1}^{(n+1,L)}) - T_{A_2}u_{A_2}^{(n+1,\ell)})\right), \\ \ell = 0, ..., M-1, \end{cases} \\ u^{(n+1)} := u_{A_1}^{(n+1,L)} + u_{A_2}^{(n+1,M)}. \end{cases} \tag{2.18}$$

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 (2.14) is proven [39]. Nothing is known about the rate of convergence, which is still an open problem, however the great advantages of this subspace correction algorithm are the fact that we can solve several smaller problems, instead of a large one. This may lead us to an acceleration of convergence due to preconditioning effects with a reduction of overall computational cost.

2.1.5 Multidomain Splitting

The above described subspace correction algorithm is not restricted to a decomposition into two subspaces, but can be generalized to an algorithm for multiple decompositions. We split now the domain into multiple disjoint sets $A_i, i = 1, 2, ..., N$,such that $A = \cup_{i=1}^{N} A_i$. Associated with this decomposition we define $V_i = \{A_2 \in \ell_2(A) : supp(u_A) \subset A_i\}$ for $i = 1, 2, ..., N$. Then we minimize J by using the following alternating algorithm: take an initial $u^{(0)} = u_{A_1}^{(0,L)} + ... + u_{A_N}^{(0,L_N)} \in V_1 \oplus V_2 .. \oplus V_N$, for example, $u^{(0)} = 0$, and iterate

$$
\begin{cases}
\begin{cases}
u_{A_1}^{(n+1,0)} = u_{A_1}^{(n,L_1)}, \\
u_{A_1}^{(n+1,\ell+1)} = S_\alpha \left(u_{A_1}^{(n+1,\ell)} + T_{A_1}^{*}((g - \sum_{i=2}^{N} T_{A_i} u_{A_i}^{(n,Li)}) - T_{A_1} u_{A_1}^{(n+1,\ell)}) \right), \\
\ell = 0, ..., L_1 - 1,
\end{cases} \\
... \\
\begin{cases}
u_{A_N}^{(n+1,0)} = u(_{A_N}^{n,L_N}), \\
u_{A_N}^{(n+1,\ell+1)} = S_\alpha \left(u_{A_N}^{(n+1,\ell)} + T_{A_N}^{*}((g - \sum_{i=1}^{N-1} T_{A_N} u_{A_N}^{(n+1,Li)} i) - T_{A_N} u^{(n+1,\ell)}) \right), \\
\ell = 0, ..., L_N - 1,
\end{cases} \\
u^{(n+1)} := \frac{\sum_{i=1}^{N} u_{Ai}^{(n+1,Li)} + (N-1)u^{(n)}}{N}.
\end{cases}
..
$$

The monotonicity of the energy with respect to the iterations and the convergence to an expected minimizer is ensured by *M.Fornasier* et Al in [39].

2.2 DOMAIN DECOMPOSITION FOR TOTAL VARIATION MINIMIZATION

This section is dedicated to overlapping and non-overlapping domain decomposition methods for total variation minimization. In order to successfully show convergence of these methods. The subminimization problems in the overlapping and non-overlapping domain decomposition methods are solved by the iterative oblique thresholding, which is based on an iterative proximity map algorithm and the computation of a Lagrange multiplier by a fixed point iteration.

2.2.1 The Overlapping Domain Decomposition Algorithm

We are interested in the minimization of the functional

$$J(u) := \|Tu - g\|_2^2 + 2\alpha|\nabla(u)|(\Omega), \tag{2.19}$$

where $T : H \to H$ is now any bounded linear operator, $g \in H$ is a given data, and $\alpha > 0$, is a fixed constant. We recall that in order to guarantee the existence of minimizers for (2.19) **we assume condition** (C), i.e., that J is coercive in H. Now, instead of minimizing (2.19) 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.,

$$\begin{aligned} |\nabla u|(\Omega) &= |\nabla u_{|\Omega_1}|(\Omega_1) + c_1(u_{|(\Omega_2\setminus\Omega_1)\cup\Gamma_1}), \\ |\nabla u|(\Omega) &= |\nabla u_{|\Omega 2}|(\Omega 2) + c_2(u_{|(\Omega_1\setminus\Omega_2)\cup\Gamma_2}), \end{aligned} \tag{2.20}$$

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. The simplest examples of discrete domains with such a property are discrete d-dimensional rectangles (d-orthotopes). For instance, with our notations, 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 (2.20) 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 H is split into two closed subspaces $Vj = \{u \in H : supp(u) \subset \Omega_j\}$, for $j = 1, 2$. Note that $H = V_1 + V_2$ is not a direct sum

of V_1 and V_2, but just a linear sum of subspaces. Thus any $u \in H$ has a non unique representation

$$u(x) = \begin{cases} u_1(x) & x \in \Omega_1 \backslash \Omega_2, \\ u_1(x) + u_2(x) & x \in \Omega_1 \cap \Omega_2, \quad u_i \in V_i, i = 1, 2. \\ u_2(x) & x \in \Omega_2 \backslash \Omega_1, \end{cases} \tag{2.21}$$

We denote by Γ_1 the interface between Ω_1 and $\Omega_2 \backslash \Omega_1$ and by Γ_2 the interface between Ω_2 and $\Omega_1 \backslash \Omega_2$ (the interfaces are naturally defined in the discrete setting). We introduce the trace operator of the restriction to a boundary Γ_i

$$Tr_{|\Gamma i} : V_i \to \mathbb{R}^{\Gamma_i}, \qquad i = 1, 2$$

with $Tr_{|\Gamma i}(v_i) = v_{i|\Gamma_i}$, 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 abounded uniform partition of unity $\{\chi 1, \chi 2\} \subset H$ such that

$(a) T_{r|\Gamma_i} \chi_i = 0$ for $i = 1, 2$,

$(b) \chi_1 + \chi_2 = 1$,

(c)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

$$\arg\min_{u \in H} J(u) \tag{2.22}$$

by taking an initial $u^{(0)} = \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} \in V_1 + V_2$,e.g., $\tilde{u}_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} u_1^{(n+1)} \approx \arg\min_{\substack{v_1 \in V_1 \\ T_{r|\Gamma_1} v_1 = 0}} J(v_1 + \tilde{u}_2^{(n)}), \\ u_2^{(n+1)} \approx \arg\min_{\substack{v_2 \in V_2 \\ T_{r|\tilde{u}_2} v_2 = 0}} J(u_1^{(n+1)} + v_2), \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)}, \\ \tilde{u}_1^{(n+1)} := \chi_1 \cdot u^{(n+1)}, \\ \tilde{u}_2^{(n+1)} := \chi_2 \cdot u^{(n+1)}. \end{cases} \tag{2.23}$$

Note that we are minimizing over functions $v_i \in V_i$ for $i = 1, 2$ that vanish on the interior boundaries, i.e., $T_{r|\Gamma_i} v_i = 0$. 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 (27)*. 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 $u_1^{(n)} = \tilde{u}_1^{(n)}$and $u_2^{(n)} = \tilde{u}_2^{(n)}$. The realization of the approximate solution to the individual subspace minimizations, discussed in the next section, for the general subspace correction algorithm in (3.5).

2.2.2 Local Minimization by Lagrange Multipliers

Let us consider, for example, the subspace minimization on Ω_1

$$\arg \min_{\substack{v_1 \in V_1 \\ T_{r|_{\Gamma_1}} v_1 = 0}} J(v_1 + u_2) = \arg \min_{\substack{v_1 \in V_1 \\ T_{r|_{\Gamma_1}} v_1 = 0}} \|Tv_1 - (g - Tu_2)\|_2^2 + 2\alpha|\nabla(v_1 + u_2)|(\Omega). \tag{2.24}$$

We observe that $\left\{u \in H : T_{r|_{\Gamma_1}} u = T_{r|_{\Gamma_1}} u_2, J(u) \leq C\right\} \subset \{J \leq C\}$. By assumption (C) these sets are bounded and hence the minimization problem (2.24) has solutions.

In order to realize an approximate solution to (2.24) we use the following algorithm: for $u_1^{(0)} = \tilde{u}_1^{(0)} \in V_1$,

$$u_1^{(\ell+1)} = \arg \min_{\substack{v_1 \in V_1 \\ T_{r|_{\Gamma_1}} v_1 = 0}} J_1^s(u_1 + u_2, u_1^{(\ell)}), \ell \geq 0, \tag{2.25}$$

where J_1^s is the surrogate functional of J defined , i.e., for a, $u_1 \in V_1$, $u_2 \in V_2$ we have

$$J_1^s(u_1 + u_2, a) = J(u_1 + u_2) + \|u_1 - a\|_2^2 - \|T(u_1 - a)\|_2^2. \tag{2.26}$$

Note that J_1^s can be written in the following form

$$J_1^s(u_1 + u_2, a) = \left\|u_1 - (a + (T * (g - Tu_2 - Ta))_{|\Omega_1})\right\|_2^2 + 2\alpha|\nabla(u_1 + u_2)|(\Omega) + \Phi(a, g, u_2),$$

with Φ being a function of a, g, u_2 only. Additionally in (2.25) we can restrict the total variation on Ω_1 only, since we have

$$|\nabla(u_1+u_2)|(\Omega) = \left|\nabla(u_1+u_2)_{|\Omega_1}\right|(\Omega_1) + c_1(u_2|_{(\Omega_2\setminus\Omega_1)\cup\Gamma_1)}, \tag{2.27}$$

where we used (2.20) and the assumption that u_1 vanishes on the interior boundary Γ_1. Hence (2.25) is equivalent to

$$\arg\min_{\substack{u_1\in V_1\\ T_{r|_{\Gamma_1}}u_1=0}} J_1^s(u_1+u_2,u_1^{(\ell)}) = \arg\min_{\substack{u_1\in V_1\\ T_{r|_{\Gamma_1}}u_1=0}} \|u_1-z_1\|_2^2 + 2\alpha\,|\nabla(u_1+u_2)|\Omega_1|\,(\Omega_1), \tag{2.28}$$

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 (2.28) we have to solve a constrained minimization problem, i.e.,

$$\arg\min_{x\in H}\{F(x) : Gx = 0\}, \tag{2.29}$$

where $F : H \to \mathbb{R}$ is a convex functional and $G : H \to H$ is a bounded linear operator on H. We have the following useful result,

Theorem 22. . *([47]). Let $N = \{G^*\lambda : \lambda \in H\} = Range(G^*)$. Then, $x^0 \in \{x \in H : Gx = 0\}$ solves the constrained minimization problem (2.29) if and only if*

$$0 \in \partial F(x_0) + N. \tag{2.30}$$

Oblique Thresholding (OT)

We want to exploit *theorem (22)* in order to produce an algorithmic solution to each iteration step (2.25), which practically stems from the solution of a problem of this type

$$\arg\min_{\substack{u_1\in V_1\\ T_{r|_{\Gamma_1}}u_1=0}} \|u_1-z_1\|_2^2 + 2\alpha\left|\nabla(u_1+u_2)_{|\Omega_1}\right|(\Omega_1).$$

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 what follows all the involved quantities are restricted to Ω_1, e.g.,$u_2 = u_{2|\Omega_1}$.

Theorem 23. *(Oblique thresholding). For $u_2 \in V_2$ and for $z_1 \in V_1$ the following statements are equivalent:*

(i) $u_1^* = \arg\min_{\substack{u_1 \in V_1 \\ Tr_{|\Gamma_1} u_1 = 0}} \|u_1 - z_1\|_2^2 + 2\alpha \left|\nabla(u_1 + u_2)\right|(\Omega_1)$;

(ii) there exists $\eta \in Range(Tr_{|\Gamma_1})^* = \{\eta \in V_1 \text{ with } supp(\eta) = \Gamma_1\}$ such that

$0 \in u_1^* - (z_1 - \eta) + \alpha \partial V_1 |\nabla(\cdot + u_2)|(\Omega_1)(u_1^*)$;

(iii) there exists $\eta \in V_1$with $supp(\eta) = \Gamma_1$ such that $u_1^* = (I - P_{\alpha K})(z_1 + u_2 - \eta) - u_2 \in V_1$ and $Tr_{|\Gamma_1}\ u_1^* = 0$;

(iv) there exists $\eta \in V_1$ with $supp(\eta) = \Gamma_1$ such that $Tr_{|\Gamma_1}\eta = Tr_{|\Gamma_1} z_1 + Tr_{|\Gamma_1} P\alpha K(\eta - (z_1 + u_2))$ or equivalently

$$\eta = (Tr_{|\Gamma_1})^* Tr_{|\Gamma_1}(z_1 + P_{\alpha K}(\eta - (z_1 + u_2))). \tag{2.31}$$

The proof follows analogue arguments as the one of Theorem 3.1.4 in [47] by just correctly replacing the projection π_{V_2} by the trace operator $Tr_{|\Gamma_1}$ and by replacing the spacesV_i with the new ones respectively.

Proposition 24. *The following statements are equivalent:*

(i) there exists $\eta \in V_1$ such that $\eta = (Tr_{|\Gamma_1})^* Tr_{|\Gamma_1}(z_1 + P_{\alpha K}(\eta - (z_1 + u_2)))$ (which is in turn the condition (iv) of Theorem (23)

(ii) the sequence $(\eta^{(m)})_m$ produced by the following iterative algorithm

$$\begin{aligned} \eta^{(0)} &\in V_1, supp\ \eta^{(0)} = \Gamma_1 \\ \eta^{(m+1)} &= (Tr_{|\Gamma_1}) * Tr_{|\Gamma_1}\left(z_1 + P_{\alpha K}(\eta^{(m)} - (z_1 + u_2))\right), m \geq 0. \end{aligned} \tag{2.32}$$

converges to any $\eta \in V_1$ that satisfies (2.31).

Convergence of the subspace minimization

From the results of the previous section it follows that the iteration (2.25) can be explicitly computed by

$$u_1^{(\ell+1)} = S\alpha(u_1^{(\ell)} + T^*(g - Tu_2 - Tu_1^{(\ell)}) + u_2 - \eta^{(\ell)}) - u_2, \tag{2.33}$$

where $S_\alpha := I - P_\alpha K$ and $\eta^{(\ell)} \in V_1$ is any solution of the fixed point equation

$$\eta = (Tr_{|\Gamma_1})^* Tr_{|\Gamma_1} \left((u_1^{(\ell)} + T^*(g - Tu_2 - Tu_1^{(\ell)})) - P_{\alpha K}(u_1^{(\ell)} + T^*(g - Tu_2 - Tu_1^{(\ell)} + u_2 - \eta)) \right).$$

The computation of $\eta^{(\ell)}$ can be implemented by the algorithm in (2.32).

Proposition 25. *Assume $u_2 \in V_2$ and $\|T\| < 1$. Then the iteration (2.33) converges to a solution $u_1^* \in V_1$ of (2.24) 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. in [47] 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.

2.2.3 Convergence of the Sequential Domain Decomposition Method

In this subsection we want to prove the convergence of the algorithm in (2.23) to minimizers of J. In order to do that, we need a characterization of solutions of the minimization problem (2.22) as the one provided in [[61], Proposition 4.1] for the continuous setting and specified for the discrete setting in Proposition 3.2.2 in [47].

Convergence properties

We return to the sequential algorithm in (2.23). Let us explicitly express the algorithm as follows: take an initial $u^{(0)} = \tilde{u}_1^{(0)} + \tilde{u}_2^{(0)} \in V_1 + V_2$, for example, $\tilde{u}_i^{(0)} = 0, i = 1, 2$, and

iterate

$$
\begin{cases}
\begin{cases} u_1^{(n+1,0)} = \tilde{u}_1^{(n)}, \\ u_1^{(n+1,\ell+1)} = \arg\min\limits_{\substack{u_1 \in V_1 \\ Tr_{|\Gamma_1} u_1 = 0}} J_1^s(u_1 + \tilde{u}_2^{(n)}, u_1^{(n+1,\ell)}) \quad \ell = 0, ..., L-1, \end{cases} \\
\begin{cases} u_2^{(n+1,0)} = \tilde{u}_2^{(n)}, \\ u_2^{(n+1,m+1)} = \arg\min\limits_{\substack{u_2 \in V_2 \\ Tr_{|\Gamma_2} u_2 = 0}} J_2^s(u_1^{(n+1,L)} + u_2, u_2^{(n+1,m)}) \quad m = 0, ..., M-1, \end{cases} \\
u^{(n+1)} := u_1^{(n+1,L)} + u_2^{(n+1,M)}, \\
\tilde{u}_1^{(n+1)} := \chi_1 \cdot u^{(n+1)}, \\
\tilde{u}_2^{(n+1)} := \chi_2 \cdot u^{(n+1)}.
\end{cases} \tag{2.34}
$$

The algorithm in (2.34) consists of two nested iterations. The inner iterations with indexes ℓ and m constitute the iterative solution for the sequence of surrogate function on each subspace. Hence, these iterations approximatively compute minimizers for the functional J on the subspaces. The outer iteration with index n stems from our domain decomposition approach and iteratively computes the minimizer of 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 $u_i^{(n+1)} = \tilde{u}_i^{(n+1)}$, $i = 1, 2$, in general. In this section we want to prove the convergence of the algorithm in 2.34 for any choice of L and M.

Proposition 26. *(Convergence properties). Let us assume that $\|T\| < 1$. The algorithm in 2.34 produces a sequence $(u^{(n)})_n$ in H with the following properties:*

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

(ii)$\lim_{n\to\infty} \left\|u^{(n+1)} - u^{(n)}\right\|_2 = 0$;

(iii) the sequence $(u^{(n)})_n$ has subsequences that converge in H.

We will skip the proof of this proposition, since it follows analogue arguments as the one of Theorem 3.1.12 in [47], .

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 27. *The sequences $(\tilde{u}_1^{(n)})_n$ and $(\tilde{u}_2^{(n)})_n$ produced by the algorithm in (2.34) are bounded, i.e., there exists a constant $\tilde{C} > 0$ such that $\left\|\tilde{u}_i^{(n)}\right\|_2 \leq \tilde{C}$ for $i = 1, 2$.*

Proof. From the boundness of $(u^{(n)})_n$ we have

$$\begin{aligned} \left\|\tilde{u}_i^{(n)}\right\|_2 &= \left\|\chi_i u^{(n)}\right\|_2 \leq c_\chi \left\|u^{(n)}\right\|_2 \leq \tilde{C}. \\ \text{for} \quad i &= 1, 2 \end{aligned}$$

Lemma 28. *The sequences $(\eta_1^{(n,L)})_n$ and $(\eta_2^{(n,M)})_n$ are bounded.*

Proof. From previous considerations we know that

$$\begin{aligned} u_1^{(n,L)} &= S_\alpha(z_1^{(n,L-1)} + \tilde{u}_2^{(n-1)} - \eta_1^{(n,L)}) - \tilde{u}_2^{(n-1)}, \\ u_2^{(n,M)} &= S_\alpha(z_2^{(n,M-1)} + u_1^{(n,L)} - \eta_2^{(n,M)}) - u_1^{(n,L)}. \end{aligned}$$

Assume that $(\eta_1^{(n,L)})_n$ were unbounded, (iii), also $S_\alpha(z_1^{(n,L-1)} + \tilde{u}_2^{(n-1)} - \eta_1^{(n,L)})$ would be unbounded. By the monotonicity property of J, see *proposition (26)*, we obtain :

$$\sum_{\ell=0}^{L-1} \left\|u_1^{(n+1,\ell+1)} - u_1^{(n+1,\ell)}\right\|_2^2 + \sum_{m=0}^{M-1} \left\|u_2^{(n+1,m+1)} - u_2^{(n+1,m)}\right\|_2^2 \to 0, n \to \infty. \tag{2.35}$$

Since $(\tilde{u}_2^{(n)})_n$ and $(u_1^{(n,L)})_n$ are bounded by *Lemma (27)* and formula (2.35), 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 (2.34) is indeed converging to a minimizer of the original functional J.

Theorem 29. *(Convergence to minimizers). Assume $\|T\| < 1$. Then accumulation points of the sequence $(u^{(n)})_n$ produced by the algorithm in (2.34) are minimizers of J. If 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 (27)* 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

$$u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)} = \tilde{u}_1^{(\infty)} + \tilde{u}_2^{(\infty)}, \tag{2.36}$$

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 (2.35) it directly follows that

$$\left\| u_2^{(n+1,M)} - \tilde{u}_2^{(n)} \right\|_2 \to 0, n \to \infty. \tag{2.37}$$

Moreover, since $\chi_2 \in V_2$ is a fixed vector which is independent of n, we obtain from *Proposition (26)* (ii) that

$$\left\| \chi_2 (u^{(n)} - u^{(n+1)}) \right\|_2 \to 0, n \to \infty,$$

and hence

$$\left\| \tilde{u}_2^{(n)} - \tilde{u}_2^{(n+1)} \right\|_2 \to 0, n \to \infty. \tag{2.38}$$

Putting (2.37) and (2.38) together and noting that

$$\left\| u_2^{(n+1,M)} - \tilde{u}_2^{(n)} \right\|_2 + \left\| \tilde{u}_2^{(n)} - \tilde{u}_2^{(n+1)} \right\|_2 \geq \left\| u_2^{(n+1,M)} - \tilde{u}_2^{(n+1)} \right\|_2$$

we have

$$\left\| u_2^{(n+1,M)} - \tilde{u}_2^{(n+1)} \right\|_2 \to 0, n \to \infty, \tag{2.39}$$

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 can denote by $u_2^{(\infty)}$. Then from (2.39) and (2.36) it directly follows that $\tilde{u}_1^{(\infty)} = u_1^{(\infty)}$.

We set

$$F_1(u_1^{(n+1,L)}) := \left\| u_1^{(n+1,L)} - z_1^{(n+1,L)} \right\|_2^2 + 2\alpha |\nabla (u_1^{(n+1,L)} + \tilde{u}_{2|\Omega_1}^{(n)})|(\Omega_1),$$

where

$$z_1^{(n+1,L)} := u_1^{(n+1,L-1)} + \left(T^*(g - T\tilde{u}_2^{(n)} - Tu_1^{(n+1,L-1)})\right)_{|\Omega_1}$$

The optimality condition for $u_1^{(n+1,L)}$ is

$$0 \in \partial_{V_1} F(u_1^{(n+1,L)}) + 2\eta_1^{(n+1,L)}$$

where

$$\eta_1^{(n+1,L)} = (Tr_{|\Gamma_1}) * Tr_{|\Gamma_1}\left((z_1^{(n+1,L)}) + P_{\alpha K}(\eta_1^{(n+1,L)} - z_1^{(n+1,L)} - \tilde{u}_2^{(n)})\right).$$

In order to use the characterization of elements in the subdifferential of $|\nabla u|(\Omega)$, i.e., (Proposition 3.2.2 in [47]), we have to rewrite the minimization problem for F_1. More precisely, we define

$$\hat{F}_1(\xi_1^{(n+1,L)}) := \left\|\xi_1^{(n+1,L)} - \tilde{u}_{2|\Omega_1}^{(n)} - z_1^{(n+1,L)}\right\|_2^2 + 2\alpha|\nabla(\xi_1^{(n+1,L)})|(\Omega_1)$$

for $\xi_1^{(n+1,L)} \in V_1$ with $Tr_{|\Gamma_1}\xi_1^{(n+1,L)} = \tilde{u}_2^{(n)}$. Then the optimality condition for $\xi_1^{(n+1,L)}$ is

$$0 \in \partial\hat{F}_1(\xi_1^{(n+1,L)}) + 2\eta_1^{(n+1,L)}. \tag{2.40}$$

Note that indeed $\xi_1^{(n+1,L)}$ is optimal if and only if $u_1^{(n+1,L)} = \xi_1^{(n+1,L)} - \tilde{u}_{2|\Omega_1}^{(n)}$ is optimal.

Analogously we define

$$\hat{F}_2(\xi_2^{(n+1,M)}) := \left\|\xi_2^{(n+1,M)} - u_{1|\Omega_2}^{(n+1,L)} - z_2^{(n+1,M)}\right\|_2^2 + 2\alpha|\nabla(\xi_2^{(n+1,M)})|(\Omega_2)$$

for $\xi_2^{(n+1,M)} \in V_2$ with $Tr_{|\Gamma_2}\xi_2^{(n+1,M)} = u_1^{(n+1,L)}$, and the optimality condition for $\xi_2^{(n+1,M)}$ is

$$0 \in \partial\hat{F}_2(\xi_2^{(n+1,M)}) + 2\eta_2^{(n+1,M)}, \tag{2.41}$$

where

$$\eta_2^{(n+1,M)} = (Tr_{|\Gamma_2}) * Tr_{|\Gamma_2}\left((z_2^{(n+1,M)}) + P_{\alpha K}(\eta_2^{(n+1,M)} - z_2^{(n+1,M)} - u_1^{(n+1,L)})\right).$$

Let us recall that now we are considering functionals with $\varphi(s) = s$, $T = I$, and $\Omega = \Omega_i, i = 1, 2$. 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 $M_1^{(n+1)} = (M_{0,1}^{(n+1)}, \overline{M}_1^{(n+1)}) \in V_1 \times V_1^d$ with $|\overline{M}_1^{(n+1)}(x)| \leq 2\alpha$ for all $x \in \Omega_1$ such that

$$\left\langle \overline{M}_1^{(n+1)}(x), (\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}))(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n)}))(x)| = 0 \tag{2.42}$$

$$-2(u_1^{(n+1,L)}(x) - z_1^{(n+1,L)}(x)) - \operatorname{div} \overline{M}_1^{(n+1)}(x) - 2\eta_1^{(n+1,L)}(x) = 0, \tag{2.43}$$

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., $-2\eta_2^{(n+1,M)} \in \partial \widehat{F}_2(\xi_2^{(n+1,M)})$, if and only if there exists an $M_2^{(n+1)} = (M_{0,2}^{(n+1)}, \overline{M}_2^{(n+1)}) \in V_2 \times V_2^d$ with $|\overline{M}_2^{(n+1)}(x)| \leq 2\alpha$ for all $x \in \Omega_2$ such that

$$\left\langle \overline{M}_2^{(n+1)}(x), (\nabla(u_1^{(n+1,L)} + u_2^{(n+1,M)}))(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(n+1,L)} + \tilde{u}_2^{(n+1,M)}))(x)| = 0 \tag{2.44}$$

$$-2(u_2^{(n+1,M)}(x) - z_2^{(n+1,M)}(x)) - \operatorname{div} \overline{M}_2^{(n+1)}(x) - 2\eta_2^{(n+1,M)}(x) = 0, \tag{2.45}$$

for all $x \in \Omega_2$. Since $(\overline{M}_1^{(n)}(x))_n$ is bounded for all $x \in \Omega_1$ and $(\overline{M}_2^{(n)}(x))_n$ is bounded for all $x \in \Omega_2$, there exist convergent subsequences $(\overline{M}_1^{(n_k)}(x))_k$ and $(\overline{M}_2^{(n_k)}(x))_k$. Let us denote $\overline{M}_1^{(\infty)}(x)$ and $\overline{M}_2^{(\infty)}(x)$ the respective limits of the sequences. For simplicity we rename such sequences by $(\overline{M}_1^{(n)}(x))_n$ and $(\overline{M}_2^{(n)}(x))_n$.

Note that, by *Lemma (28)*(or simply from (2.43) and (2.45) 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 (2.42)-(2.45) the limits for $n \to \infty$ we obtain

$$\begin{aligned} \left\langle \overline{M}_1^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| &= 0 \text{ for all } x \in \Omega_1, \\ -2(u_1^{(\infty)}(x) - z_1^{(\infty)}(x)) - div\overline{M}_1^{(\infty)}(x) - 2\eta_1^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega_1 \end{aligned}$$

$$\begin{aligned}\left\langle \overline{M}_2^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| &= 0 \text{ for all } x \in \Omega_2, \\ -2(u_2^{(\infty)}(x) - z_2^{(\infty)}(x)) - div\overline{M}_2^{(\infty)}(x) - 2\eta_2^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega_2\end{aligned}$$

Since $supp\ \eta_1^{(\infty)} = \Gamma_1$ and $supp\ \eta_2^{(\infty)} = \Gamma_2$ we have

$$\begin{aligned}\left\langle \overline{M}_1^{(\infty)}(x), (\nabla(u^{(\infty)})(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \text{ for all } x \in \Omega_1, \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - div\overline{M}_1^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega_1 \backslash \Gamma_1\end{aligned} \tag{2.46}$$

$$\begin{aligned}\left\langle \overline{M}_2^{(\infty)}(x), (\nabla(u^{(\infty)})(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \text{ for all } x \in \Omega_2, \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - div\overline{M}_2^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega_2 \backslash \Gamma_2.\end{aligned} \tag{2.47}$$

Observe now that from (proposition 3.2.2 in [47]) we also have that $0 \in J(u^{(\infty)})$ if and only if there exists $M^{(\infty)} = (M_0^{(\infty)}\ , \overline{M}^{(\infty)})$ with $|\overline{M}^{(\infty)}(x)| \leq 2\alpha$ for all $x \in \Omega$ such that

$$\begin{aligned}\left\langle \overline{M}^{(\infty)}(x), (\nabla(u^{(\infty)})(x) \right\rangle \mathbb{R}^d + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \text{ for all } x \in \Omega, \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - div\overline{M}^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega.\end{aligned} \tag{2.48}$$

Note that $\overline{M}_j^{(\infty)}(x), j = 1, 2$, for $x \in \Omega_1 \cap \Omega_2$ satisfies both (2.46) and (2.47) .

Hence let us choose

$$M^{(\infty)}(x) = \begin{cases} M_1^{(\infty)}(x) \text{ if } x \in \Omega_1 \backslash \Gamma_1, \\ M_2^{(\infty)}(x) \text{ if } x \in (\Omega_2 \backslash \Omega_1) \cup \Gamma_1. \end{cases}$$

With this choice of $M^{(\infty)}$ the equations (2.46) - (2.48) are valid and hence $u^{(\infty)}$ is optimal in Ω.

Remark 30. *(i) If* $\nabla u^{(\infty)}(x) = 0$ *for* $x \in \Omega_j, j = 1, 2$, *then* $\overline{M}_j^{(\infty)}$ *is given by*

$$\overline{M}_j^{(\infty)}(x) = -2\alpha \frac{(\nabla u_{|\Omega_j}^{(\infty)})(x)}{|(\nabla u_{|\Omega_j}^{(\infty)})(x)|}$$

(ii) The boundedness of the sequences $(\widetilde{u}_1^{(n)})_n$ *and* $(\widetilde{u}_2^{(n)})_n$ *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 (29). *Their boundedness is guaranteed as in* Lemma (27) *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 (2.48) for $u^{(\infty)}$ *we combined the respective conditions (2.46) and (2.47) 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 31. *The generalization of the algorithm to a multiple domain decomposition is straightforward. Let us split now* Ω *into* $N \geq 2$ *overlapping domains* $\Omega_i, i = 1, ..., N$. *Associated with this decomposition we define* $V_i := \{u \in H : supp(u) \subset \Omega_i\}$ *such that* $H = V_1 + ... + V_N$ *and we denote* $u_i = \pi_{V_i} u$ *for* $i = 1, ..., N$. *By* $\Gamma_i = \partial\Omega_i \backslash \partial\Omega$ *we denote the inner interfaces of the domain patches. Further we fix a bounded uniform partition of unity* $(BUPU)\{\chi 1, ..., \chi N\} \subset H$ *such that*

(a) $Tr_{|\Gamma_i} \chi_i = 0$ *for* $i = 1, ..., N$,

(b) $\sum_{i=1}^N \chi_i = 1$,

(c) *supp* $\chi_i \subset \Omega_i$ for $i = 1, ..., N$,

(d) $\max\{\|\chi_1\|_\infty, ..., \|\chi_N\|_\infty\} = c_\chi < \infty$.

Then we define the overlapping multiple domain decomposition algorithm as follows: for an initial $V_1 + ... + V_N \ni \widetilde{u}_1^{(0)} + ... + \widetilde{u}_N^{(0)} := u^{(0)} \in H$, for example, $\widetilde{u}_i^{(0)} = 0$, $i = 1, ..., N$, use

the iteration

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = \widetilde{u}_1^{(n)}, \\ u_1^{(n+1,\ell+1)} = \arg\min_{\substack{u_1 \in V_1 \\ Tr_{|\Gamma_1} u_1 = 0}} J_1^s(u_1 + \sum_{i=2}^N \widetilde{u}_i^{(n)}, u_1^{(n+1,\ell)}), \\ \ell = 0, ..., L_1 - 1, \end{cases} \\ \qquad \ldots \\ \begin{cases} u(n+1,0)N = \widetilde{u}_N^{(n)}, \\ u_N^{(n+1,\ell+1)} = \arg\min_{\substack{u_N \in V_N \\ Tr_{|\Gamma_N} u_N = 0}} J_N^s(\sum_{i=1}^{N-1} u_i^{(n+1,L)} + u_N, u_N^{(n+1,\ell)}), \\ \ell = 0, ..., L_N - 1, \end{cases} \\ u^{(n+1)} := \sum_{i=1}^N u_i^{(n+1,L)}, \\ \widetilde{u}_i^{(n+1)} := \chi_i \cdot u^{(n+1)} \text{ for } i = 1, ..., N. \end{cases} \tag{2.49}$$

The surrogate functionals J_i^s are defined in an analogous way as above, for instance, J_1^s is given as in (2.26) by just substituting $\sum_{i=2}^N u_i^{(n)}$ for u_2 and by using the appropriate spaces. Then one can show the same convergence properties as in *proposition (26)* and *Theorem (29)*. Hence the convergence of algorithm in (2.49) to a minimizer of the original functional (2.19) is ensured.

2.2.4 Applications and Numerical Implementations

In this section we present the application of the sequential algorithm (2.34) for the minimization of J in one and two dimensions. In particular, we give a detailed explanation of the domain decompositions used in the numerics.

Numerical Results

In the following we present numerical examples for the sequential algorithm in (2.34) in two particular applications: signal interpolation 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 J, the boundedness of the iterations due to the implementation of *bounded uniform partition of unity (*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, for a general approach in regularized inverse problems, or for a discussion of the correspondence between the noise level and α in the case of total variation minimization. In Figure 2.4 and Figure 2.5 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 J in (2.19) 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 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 J for applications of algorithm in (2.34) on two subdomains and with different overlap sizes $q = 1, 5, 10, 20, 30$. The fixed points η's are computed on a small interval $\widehat{\Omega}_i, i = 1, 2$, of size 2. These results confirm the behavior of the algorithm in (2.34) as predicted by the theory; the algorithm monotonically decreases 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 *bounded uniform partition of unity* (BUPU) with in the domain decomposition algorithm. In this case, with datum g as in Figure 2.3, we chose $\alpha = 1$ and an overlap of size $q = 10$. The fixed points η's are computed on a small interval $\widehat{\Omega}_i, i = 1, 2$ respectively, of size 6.

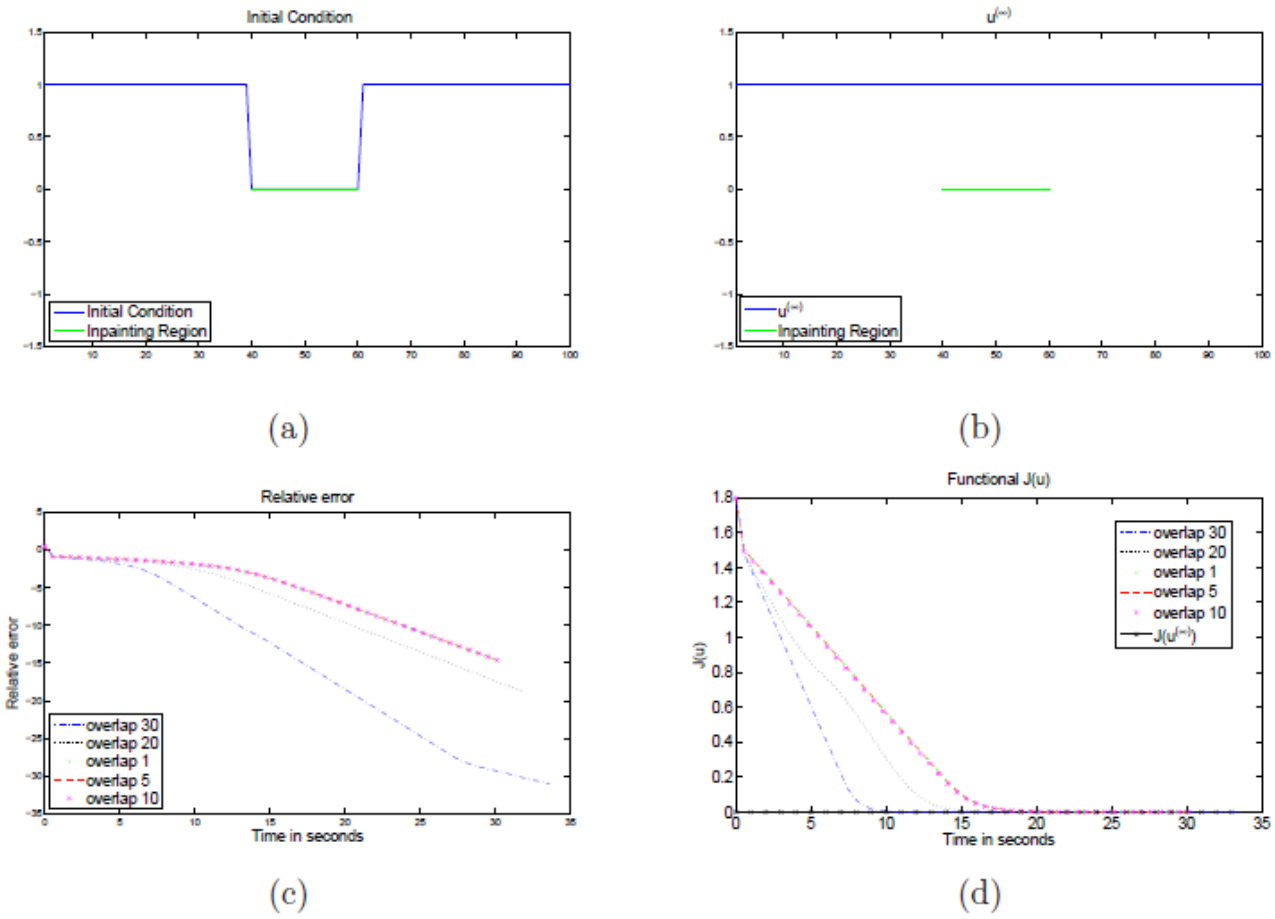

Figure 2.3. We present a numerical experiment related to the interpolation of a $1D$ signal by total variation minimization. The original signal is only provided outside of the green subinterval. The initial datumg is shown in (a). As expected, the minimizer $u^{(\infty)}$ is the constant vector 1, as shown in (b). In (c) and (d) we display the rates of decay of the relative error and of the value of J respectively, for applications of the algorithm in (2.34) with different sizes $G = 1, 5, 10, 20, 30$ of the overlapping region of two subintervals.

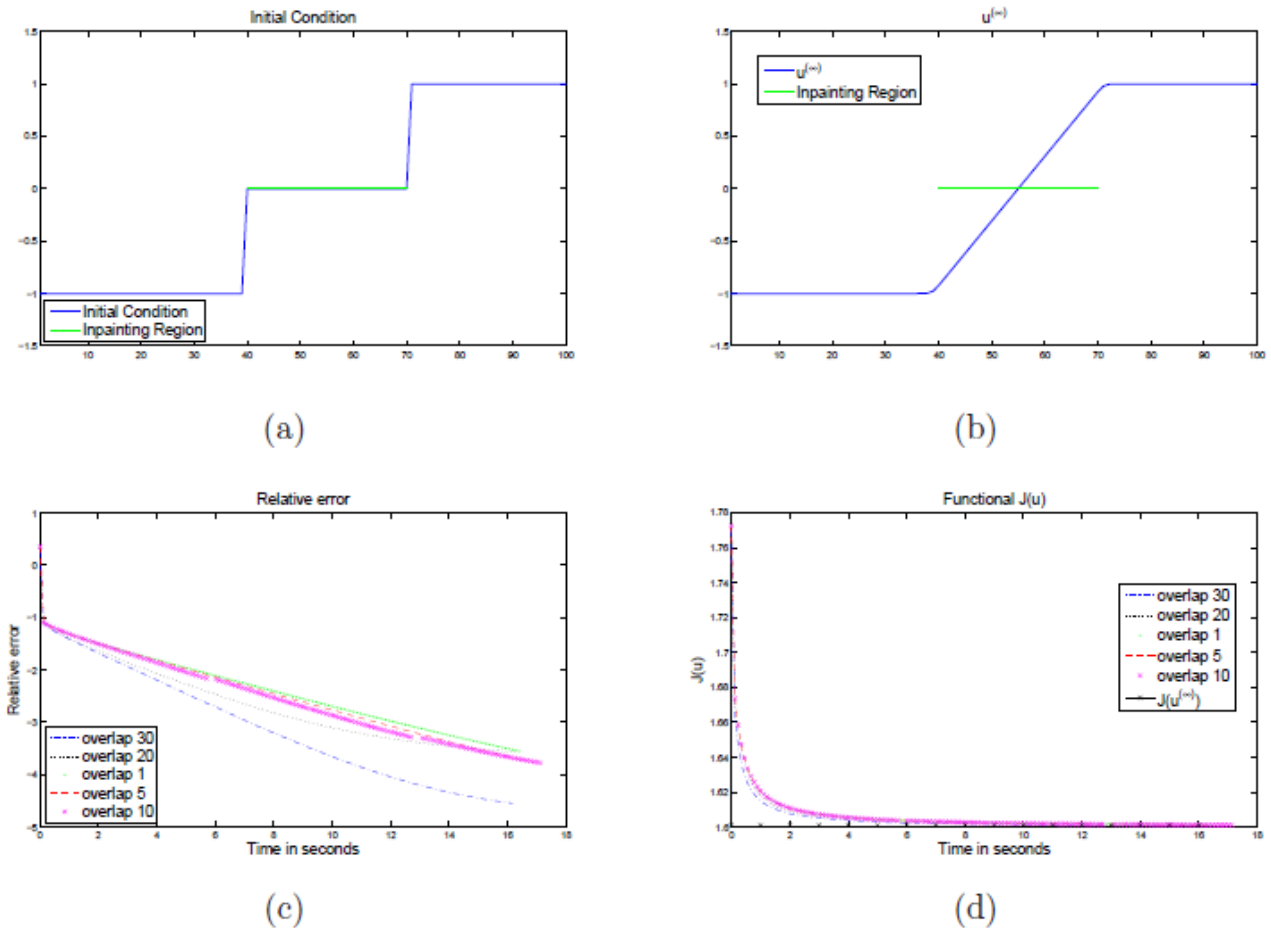

Fig. 2.4. We show a second example of total variation interpolation in $1D$. The initial datumg is shown in (a). As expected, a minimizer $u^{(\infty)}$ is (nearly) a piecewise linear function, as shown in (b). In (c) and (d) we display the rates of decay of the relative error and of the value of J respectively, for applications of the algorithm in (2.34) with different sizes $G = 1, 5, 10, 20, 30$ of the overlapping region of two subintervals.

	1 domain	4 domains	16 domains
CPU time	23086.68 s	6531.94 s	1583.52 s
Nr. outer iterations	1000	10	10

Table 2.1: 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 $\left\|u^{(n)} - u_{org}\right\|_2^2$ gets below $\varepsilon = 0.0048$.

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 2.1.

2.3 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 [47], i.e., the algorithm in (3.13) in [47] , to the case of a non-overlapping domain decomposition as suggested in [42]. The functional of interest to be minimized is again the discrete functional J in (2.19) together with the coercivity condition (C). Now, instead of minimizing J on the whole domain, we propose to decompose Ω into disjoint and non-overlapping subdomains. We limit ourselves to split the problem into two disjoint subdomains Ω_1 and Ω_2 such that $\Omega_1 \subset \Omega$ and $\Omega_2 = \Omega \backslash \Omega_1$, but one can easily generalize the splitting to multiple subdomains. 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 H is **splitted** into two closed orthogonal and complementary subspaces $Vi = \{u \in H : supp(u) \subset \Omega_i\}$,for $i = 1, 2, i.e., H = V_1 \oplus V_2$. Note that in the following $u_i = \pi_{V_i}(u)$, for $i = 1, 2$. Now we would like to solve (2.22) by picking an initial

$V_1 \oplus V_2 \ni u_1^{(0)} + u_2^{(0)} := u^{(0)} \in H$,e.g., $u_i^{(0)} = 0, i = 1, 2$, and iterate

$$\begin{cases} u_1^{(n+1)} \approx \arg\min_{v_1 \in V_1} J(v_1 + u_2^{(n)}), \\ u_2^{(n+1)} \approx \arg\min_{v_2 \in V_2} J(u_1^{(n+1)} + v_2), \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)}. \end{cases} \tag{2.50}$$

The subspace minimization problems of the algorithm in (2.50) are solved as described in Section 3.1 in [47]. That is, for J_1^s defined as in (3.4) in [47] now with the spaces V_i from above,each subspace minimization is approximated by the surrogate functional minimization

$$u_1^{(0)} \in V_1, u_1^{(\ell+1)} = \arg\min_{v_1 \in V_1} J_1^s(u_1 + u_2, u_1^{(\ell)}), \quad \ell \geq 0.$$

(cf. (3.5) in [47]), which is then solved by Lagrange multipliers or more precisely by iterative oblique thresholding.

2.3.1 Convergence of the Sequential Domain Decomposition Method

Let us return to the sequential algorithm in (3.13) in [47] and express it explicitly for the case of a non-overlapping domain decomposition as follows: take an initial

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

$$\begin{cases} \begin{cases} u_1^{(n+1,0)} = u_1^{(n,L)}, \\ u_1^{(n+1,\ell+1)} = \arg\min_{u_1 \in V_1} J_1^s(u_1 + u_2^{(n,M)}, u_1^{(n+1,\ell)}) \quad \ell = 0, ..., L-1, \end{cases} \\ \begin{cases} u_2^{(n+1,0)} = u_2^{(n,M)}, \\ u_2^{(n+1,m+1)} = \arg\min_{u_2 \in V_2} J_2^s(u_1^{(n+1,L)} + u_2, u_2^{(n+1,m)}) \quad m = 0, ..., M-1, \end{cases} \\ u^{(n+1)} := u_1^{(n+1,L)} + u_2^{(n+1,M)}. \end{cases} \tag{2.51}$$

In this section we want to prove its convergence to a minimizer of the discrete functional J for any choice of finite numbers L and M of inner iterations. We recall that by Theorem 3.1.12 in [47] the algorithm in (2.51) decreases the energy 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 (2.19). However, in the numerical experiments shown in [42], the algorithm seems always converging robustly to the expected minimizer.

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 (2.51) indeed converges to an expected minimizer in our discrete setting. Then by following the same strategy as in the proof of *Theorem (29)* we are eventually able to prove the convergence of the algorithm in (2.51) to minimizers of J.

Theorem 32. *(Convergence to minimizers). Assume $\|T\| < 1$. Then accumulation points of the sequence $(u^{(n)})_n$ produced by the algorithm in (2.51) are minimizers of J. If 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 (2.51) 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 2.2.8* we obtain with the help of the following optimality conditions

$$\begin{aligned} \left\langle \overline{M}_1^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| &= 0 \text{ for all } x \in \Omega \\ -2(u_1^{(\infty)}(x) - z_1^{(\infty)}(x)) - div\overline{M}_1^{(\infty)}(x) - 2\eta_1^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega, \end{aligned}$$

for $u_1^{(\infty)}$ and the following optimality conditions

$$\begin{aligned} \left\langle \overline{M}_2^{(\infty)}(x), (\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla(u_1^{(\infty)} + u_2^{(\infty)}))(x)| &= 0 \text{ for all } x \in \Omega \\ -2(u_2^{(\infty)}(x) - z_2^{(\infty)}(x)) - div\overline{M}_2^{(\infty)}(x) - 2\eta_2^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega, \end{aligned}$$

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

$$\begin{aligned} \left\langle \overline{M}_1^{(\infty)}(x), (\nabla(u^{(\infty)})(x) \right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \text{ for all } x \in \Omega \\ -2\pi_{V_1} T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \overline{M}_1^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega_1 \end{aligned} \tag{2.52}$$

$$\begin{aligned}\left\langle \overline{M}_2^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \text{ for all } x \in \Omega \\ -2\pi_{V_2}T^*((Tu^{(\infty)})(x) - g(x)) - \operatorname{div} \overline{M}_2^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega_2.\end{aligned} \tag{2.53}$$

Observe now that from *proposition 3.2.2* [47] we also have that $0 \in J(u^{(\infty)})$ if and only if there exists $M^{(\infty)} = (M_0^{(\infty)}, \overline{M}^{(\infty)})$ with $|M^{(\infty)}(x)| \leq 2\alpha$ for all $x \in \Omega$ such that

$$\begin{aligned}\left\langle \overline{M}^{(\infty)}(x), (\nabla(u^{(\infty)})(x)\right\rangle_{\mathbb{R}^d} + 2\alpha|(\nabla u^{(\infty)})(x)| &= 0 \text{ for all } x \in \Omega \\ -2T^*((Tu^{(\infty)})(x) - g(x)) - div\overline{M}^{(\infty)}(x) &= 0 \text{ for all } x \in \Omega.\end{aligned} \tag{2.54}$$

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(2.52)-(2.54)are valid and hence $u^{(\infty)}$ is optimal in Ω.

Remark 33. *Note that in comparison to the proof of* Theorem (29)*, here we could not use the overlapping property of the subdomains, but we took strongly advantage of the fact that supp* $\eta_1 \subset \Omega_2$ *and supp* $\eta_2 \subset \Omega_1$*. Hence we could restrict the corresponding optimality conditions in (2.52) and (2.53) to the domain* Ω_1 *and* Ω_2 *only.*

3.0 OVERLAPPING AND NONOVERLAPPING DOMAIN DECOMPOSITION METHODS FOR IMAGE RESTORATION

3.1 INTRODUCTION

In this chapter we present some applications of domain decomposition techniques to image restoration. Here we are concerned with an overlapping and nonoverlapping domain decomposition methods for image restoration by the tixotrop model. Given the observation that natural and man-made images are characterized by extensive relatively uniform parts, one may want to help the reconstruction by imposing that the interesting solution is the one which matches the given data and has also few discontinuities localized on sets of a lower dimension. In this chapter we review both nonoverlapping and overlapping domain decomposition methods for the Tixotrop model minimization and we provide their convergence properties to global minimizers. Furthermore, we show efficiency by numerical applications in classical problems of signal and image processing.

3.2 THE MINIMIZATION PROBLEM

Assume that u is a piecewise constant function as given in [39]. The multiphases piecewise constant Tixotrop model [46] is to solve the following minimization problem:

$$\min_u F(u) = \frac{\varepsilon}{p}\int_\Omega |\nabla u|^p\, dx + g\int_\Omega |\nabla(u-u_0)|\, dx + \frac{\lambda}{2}\int_\Omega |u-u_0|^2\, dx \tag{3.1}$$

Where p can be adaptively selected based on the local gradient image features that is, away from edges, p tend to 2 to preserve edges. Therefore this new model where $p = 2$ can

effectively reduce the staircase effect in TV model whereas it can still retain the sharp edges [23]. For more details, the reader is referred to [74]

3.3 DOMAIN DECOMPOSITION BASED SUBSPACE CORRECTION METHOD

We put the method in a more general setting and start with the description of the subspace correction algorithm of [39]. Given a reflexive Banach space V and a convex, Gateaux differentiable functional $F : V \to \mathbb{R}$,we consider the minimization problem:

$$\min_{u \in V} F(u) \tag{3.2}$$

Under the notion of space correction, we first subdivide the space V into a sum of smaller subspaces:

$$V = V_1 + V_2 + \cdots + V_m, \tag{3.3}$$

in the sense that for any $v \in V$, there exists $v_j \in V_j$ such that $v = \sum_{i=1}^{m} v_j$.

Following the framework of [29] and [51] for linear problems, we solve a finite sequence of sub-minimization problems over the subspaces $(V_j)_{j=1}^{n}$:

$$\min_{e \in V_j} F(u^n + e), \tag{3.4}$$

where u^n denotes a the previous approximation. Two types of subspace correction methods based, known as the parallel subspace correction (PSC) and successive subspace correction

(SSC) method, were proposed in [31], [52]. Here, we adopt the latter, which can be described as follows:

Algorithm **SSC**. Choose an initial value $u_0 \in V$.

For $n = 0$,

while $j = 1, ..., m$ do

$$\text{Find } e_j^n \in V_j \text{ such that}$$
$$F\left(u^{n+(j-1)/m} + e_j^n\right) \leq F\left(u^{n+(j-1)/m} + v_j\right), \quad \forall v_j \in V_j \tag{3.5}$$
$$\text{set}$$
$$u^{n+j/m} = u^{n+(j-1)/m} + e_j^n$$

end

Go to next iteration for n.

As an illustrative example, we apply the algorithm to the regularized Tixotrop denoising model with the cost functional:

$$F(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 \, dxdy + \alpha \int_\Omega \sqrt{\beta + |\nabla (u - u_0)|^2} dxdy \tag{3.6}$$
$$+\frac{1}{2}\int_\Omega |u - u_0|^2 \, dxdy, \quad \alpha, \beta > 0$$

where u_0 is a given noisy image defined on $\Omega = (0,1) \times (0,1)$. Here, F is differentiable and avoiding the division by zero in the corresponding Euler-Lagrange equation:

$$\Delta u - \alpha \; div \left(\frac{\nabla (u - u_0)}{\sqrt{\beta + |\nabla (u - u_0)|^2}} \right) + u = u_0 \tag{3.7}$$

with an homogenous Neumann boundary condition $\partial u / \partial n = 0$ along the boundary. Recall that the lagged diffusivity fixed-point iteration for (3.7) is to solve the linearized equation:

$$\Delta u^{k+1} - \alpha \; div \left(\frac{\nabla \left(u^{k+1} - u_0\right)}{\sqrt{\beta + |\nabla (u^k - u_0)|^2}} \right) + u^{k+1} = u_0, \quad k = 0, 1, ..., \tag{3.8}$$

with the initial value u_0. We see that each iteration involves all the pixel values in the image domain, so it will be costly and usually the system is not well conditioned when the image's size is large. The domain decomposition based **SSC** algorithm will overcome the difficulties by breaking down the whole problem into sub-problems of much smaller size.

Firstly, we use an overlapping domain decomposition to decompose the solution space $V = H^1(\Omega)$. More precisely, we procceed by a partition of the domain Ω into m overlapping subdomains

$$\Omega = \bigcup_{j=1}^{m} \Omega_j, \qquad \Omega_j \cap \Omega_k \neq \emptyset, \quad k \neq j, \tag{3.9}$$

Clearly,each subdomain Ω_j is colored with a color j, and Ω_j consists of m_j subdomains (assumed to be "blocks" for simplicity), which are not intersected. Hence, the total number of blocks that cover Ω is :

$$M = \sum_{j=1}^{m} m_j \tag{3.10}$$

Figure 1 illustrates schematically the decomposition of Ω into four colored subdomains with 25 blocks. Based on the above domain decomposition, we decompose the space $V = H^1(\Omega)$ as the following :

$$V = \sum_{j=1}^{m} V_j, \quad V_j = H_0^1(\Omega_j), \tag{3.11}$$

where $H_0^1(\Omega_j)$ denotes the subspace of $H^1(\Omega_j)$ with zero traces on the "interior" boundaries $\partial\Omega_j \backslash \partial\Omega$. by applying the SSC algorithm to the Tixotrop-denoising model leads to an iterative method.This is to say:

Given an initial value $u_0 \in V$, Algorithm SSC leads us to get u^n from

$$\begin{cases} F\left(u^{n+\frac{j-1}{m}} + e_j^n\right) \leq F\left(u^{n+\frac{j-1}{m}} + v_j\right), \forall v_j \in V_j = H_0^1(\Omega_j) \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}} + e_j^n, \qquad 1 \leq j \leq m. \end{cases} \tag{3.12}$$

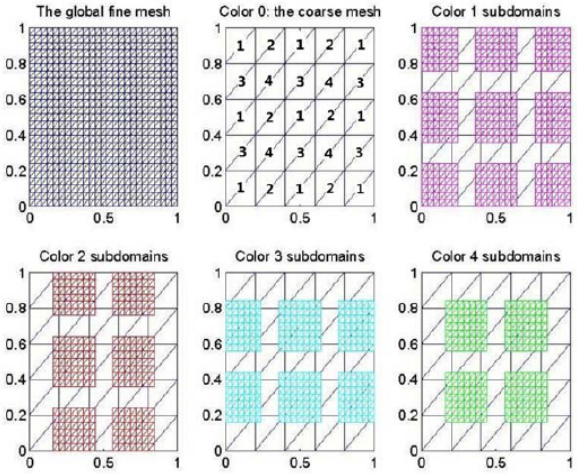

Figure 3.1. Schematic illustration of the coloring of the subdomains, and fine/coarse meshes on $\Omega = (0,1)^2$, This corresponds to the decomposition:

Here, we notice that e_j^n is the solution of the subproblem over Ω_j . It is also easy to see that $u^{n+\frac{j}{m}}$ satisfies the associated Euler-Lagrange equations for $1 \leq j \leq m$,

$$\begin{cases} \Delta u^{n+\frac{j}{m}} - \alpha \; div \left(\dfrac{\nabla\left(u^{n+\frac{j}{m}} - u_0\right)}{\sqrt{\beta + \left|\nabla\left(u^{n+\frac{j-1}{m}} - u_0\right)\right|^2}} \right) + u^{n+\frac{j}{m}} = u_0, \quad \text{in} \quad \Omega_j \\ \dfrac{\partial u^{n+\frac{j}{m}}}{\partial n} = 0, \qquad\qquad \text{on} \quad \partial\Omega_j \cap \partial\Omega, \\ u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}, \qquad\qquad \text{on} \quad \partial\Omega_j \backslash \partial\Omega. \end{cases} \tag{3.13}$$

Outside Ω_j, we have $u^{n+\frac{j}{m}} = u^{n+\frac{j-1}{m}}$ with $u^{n+\frac{j}{m}} = u(n.....)$, thus, there is no need to solve $u^{n+\frac{j-1}{m}}$ outside Ω_j. As the subdomain Ω_j may contain many disjoint "blocks", the values of $u^{n+\frac{j-1}{m}}$ can be obtained in parallel in these "blocks" by solving (3.13).

3.4 NUMERICAL ALGORITHM

At this stage, we present the full two-level algorithm formulated in the previous section for the Tixotrop denoising model.

First, we subdivide the image domain $\Omega = (0,1) \times (0,1)$ into $N \times N$ uniform cells with mesh size $h = 1/N$. The cell centers are given by

$$(x_i, y_j) = ((i-1).h, (j-1).h), \qquad 1 \leq i, j \leq N+1. \tag{3.14}$$

As a second step we procceed by letting $u^0_{i,j}$ be the pixel value of the original image u^0 at (x_i, y_j), and $u_{i,j}$ be the finite difference solution at (x_i, y_j). By using the notation:

$$\delta^{\pm}_x u_{i,j} = \pm (u_{i\pm1,j} - u_{i,j}), \qquad \delta^{\pm}_y u_{i,j} = \pm (u_{i,j\pm1} - u_{i,j}), \tag{3.15}$$

$$\delta^{c}_x u_{i,j} = (u_{i+1,j} - u_{i-1,j}), \quad \delta^{c}_y u_{i,j} = (u_{i,j+1} - u_{i,j-1}), \tag{3.16}$$

the finite difference approximation of (3.7) is:

$$u_{i,j} - \delta^+_x\left(\delta^-_x u_{i,j}\right) - \delta^+_y\left(\delta^-_y u_{i,j}\right) - \alpha_h \left\{ \begin{array}{c} \delta^-_x \left[\frac{\delta^+_x u_{i,j}}{\sqrt{\left(\delta^+_x u_{i,j}\right)^2 + \left(\delta^c_y u_{i,j}\right)^2 + \beta_h}} \right] + \\ \delta^-_y \left[\frac{\delta^+_y u_{i,j}}{\sqrt{(\delta^c_x u_{i,j})^2 + \left(\delta^+_y u_{i,j}\right)^2 + \beta_h}} \right] \end{array} \right\} = u^0_{i,j} \tag{3.17}$$

where $\alpha_h = \frac{\alpha}{h}$ and $\beta_h = 4h\beta$, with $\alpha = 0.5, \quad \beta = 0.1$.

3.5 NUMERICAL RESULTS

We present in this section various numerical results to demonstrate the efficiency of the proposed domain decomposition algorithms without and with a coarse domain correction, denoted by DD and DDC in short, respectively. If we, assume that the pixel values of all images lie in the interval $[0, 255]$, and the Gaussian white noise is added by the normal *imnoise* function *imnoise(I,'gaussian', M, σ)* (i.e., the mean M and variance σ) in Matlab. In our numerical tests, we use **PSNR** as a criteria for the quality of restoration. This quantity is usually expressed in terms of the logarithmic decibel scale by:

$$PSNR = 10 \log_{10} \frac{(255)^2}{\frac{1}{mn} \sum_{i,j} \left(u_{i,j} - u^0_{i,j}\right)^2} \tag{3.18}$$

where $\{u_{i,j} - u^0_{i,j}\}$ are the differences of the pixel values between the restored and original images. Typical values for the **PSNR** in lossy image and video compression are between *30dB* and *50dB* (the higher implies the better). Acceptable values for wireless transmission quality loss are considered to be about *20dB* to *25dB*. We shall also use the relative dynamic error between two consecutive iterations:

$$\frac{\left\|u^k - u^{k-1}\right\|_{L^2(\Omega)}}{\left\|u^k\right\|_{L^2(\Omega)}} < \varepsilon \tag{3.19}$$

for a prescribed tolerance ε, as the stopping criteria.

Figure 3.2. Row one: original image (left lena-512 $\times$ 512), noise image with $\sigma = 0.04$ (middle) and restored image(right) obtained by DD with subdomain size $d = 32$, overlapping size $\delta = 4$,$\alpha = 0.025$, $\varepsilon = \beta = 10 - 4$. Here, $PSNR = 25.9388$ (TV)

Original image (512×512)

Noise image with σ=0.04

Restored image obtained by DD with subdomain size d=32,overlapping size δ=4,α=0.025,ε=β=10-4. The PSNR=35.9388.

Figure 3.3.

Original image

Noised color image with 15% multiplicative noise.

Restored color image obtained with subdomain size $d = 64$,overlapping $size = 4$. The$PSNR = 27,303$.

Figure 3.4.

Image	d	δ	k	PSNR	Time	Image	d	δ	k	PSNR	Time
Lena 512	TV		45	25.9725	415.9900	Lena 512	Tixotrop		42	29.9782	1970.2
		1	63	25.9135	6.32%		32	1	55	29.9892	13.81%
	32	2	51	25.8780	15.32%			2	45	29.9994	12.40%
		3	46	25.8724	5.82%			3	40	29.9997	12.74%
		4	43	25.9261	16.41%			4	40	29.9697	13.96%
		5	41	25.9022	17.86%			5	39	29.9587	15.28%
		6	41	25.8859	19.84%			6	39	29.9797	16.88%
		7	45	25.9778	24.71%			7	39	29.9497	20.54%
		8	46	25.9247	28.62%			8	39	29.9897	24.10%

Table3.1.Different overlapping size with stopping residual $\varepsilon = 10^{-4}, \sigma = 0.04$

Comparison of TV, and Tixotrop, $\alpha = 0.025$ and $\beta = 10^{-4}$.

To illustrate the impact of overlapping sizes, we present results in Table 3.1 the $PSNR$ and CPU time of the classic TV by the lagged diffusivity fixed-point iteration and Tixotrop with subdomain size 32 and 64 for black and white, and color images respectively, but with different overlapping size δ. Here, the percentage of the CPU time is against TV. We see that the PSNR obtained by Tixotrop is not so sensitive to the overlapping size δ, while the computational time increases as δ increases, as expected. To have a good trade-off between the convergence rate and the quality of restoration, it is advisable to choose δ to be $2, 3$ or 4. It is essential to point out that the use of Tixotrop leads to a remarkable reduction of computational time in particular for images of large size. Numerical results are shown in Figure 3.3. for Lena-512×512 image and Figure 3.4. for a color image.

4.0 CONCLUSIONS

In this thesis, we presented a technique based on a domain decomposition. This technique involving many known results from fonctional to numerical analysis, has lead us to enumerate all the tools needed in our applications. In a preliminary chapter, we started by introducing the fonctional spaces, the conversity, the differentiability and some optimisation methods.In the second chapter we described the domain decomposition procedure in a general way. Finally, we presented our contribution resumed in a domain decomposition method for an image processing problem by the tixotrop model. As a first step, we described in details, the implementation of the domain decomposition and coarse mesh correction techniques. Then by several numerical simulations we gave useful guidelines for the choice of parameters through such quantitative studies, and demonstrate the effeciency of the implemented methods in CPU time and memory saving. The results shown in table 3.1. and Figures 3.4. and 3.5. show the advantages for our tixotrop model and the subdomain technique used. Moreover, we compared the proposed method with the dual algorithms with respect to the decay of numerical residuals. We conclude that, these techniques are very useful from many points:

1- The low cost: less CPU time is needed

2- The quality of the stored image: the PSNR is qualitatively better

3- Simpler problems are solved: a large scale problem is decomposed to subproblems of much smaller size in an efficient manner

4- Parallel computing can be used

4.1 FUTURE WORK

As an extension to this work, we plan to:

(i) implement some other algorithms for solving the Tixotrop model for vectorial and colored images.

(ii) Use parallel implementations.

4.2 APPENDIX: CODE FOR THE TIXOTROP MODEL IN MATLAB

The following is a matlab level algorithm for the solution of the tow-dimensional model problem tixotrop equation for image processing.

---

function Tixotrop_ overlap_ 2D

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Overlapping DOMAIN DECOMPOSITION FOR

Tixotrop-DENOISING AND INTERPOLATION of 2D IMAGES %%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % Domain Decomposition Method for Tixotrop Minimization.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all

t=cputime;

%choice of a grayvalue image:

image = 'Lena.png';%image = 'color image.jpg';

g=imread(image);

g=rgb2gray(g);

g=double(g);

[m,n]=size(g);

%% Definition of PARAMETERS:%%%%%%%

h=1; %spacestep size

```
dt=1/4; %timestep size
lpower=-0.5; %Lagrange multiplier \lambda_0=10^lpower
lambda0=10^lpower;
D=4; %Number of subdomains
sub=3; %number of iterations in each subdomain
erreta=10^(-9); %aimed maximal error in the computation of \eta
itetamax=10; %maximal number of iterations in the fixed point computation of \eta
projerr=10^(-2); %aimed error in the computation of the projection of Chambolle to compute the tixotrop-seminorm.
errthresh=10^(-6); %aimed error in the computation of the outer iterations, i.e. the computations of the minimizer.
overlaphalf=floor((m/D)/6);%=13
overlap=2*overlaphalf; %size of overlap
etacomp1=5;
etacomp2=0;
%%%%%%%%%%%%%%%%%%%%%
%scale image values to [0,1]
g = g./max(max(g));
%grayvalue range for plotting
clims=[0 1]
% %%%%%%%%%%%%%%%%%%%%%
%(LAMBDA = CHARACTERISTIC FUNCTION OF \OMEGA\setminus D):
lambda = ones(m,n);
% Comment the next 7 lines if used for image denoising!
[j,k] = find(g ==0); %t
sd = size(j,1);
for i=1:sd
lambda(j(i),k(i)) = 0;
end
clear j k sd
```

```
lambda = sparse(lambda);
% INITIALIZATION:
u=sparse(g);
%DOMAIN DECOMPOSITION:
%Split the problem in D subproblems with overlapping domains:
%%%%%%%%%%%%%%%%%%%%%%%%%
% % % %
% % % %
% Omega_1 %\Gamma_2 %\Gamma_1 Omega_2 %
% % % %
% % % %
%%%%%%%%%%%%%%%%%%%%%%%%%
% \Omega = \Omega_1\cup\Omega_2
% \Gamma_1 = interface of \Omega_1; \Gamma_2 = interface of \Omega_2
% u in \Omega_1\setminus\Omega_2 = u1;
% u in \Omega_1\cap\Omega_2 = u1+u2;
% u in \Omega_2\setminus\Omega_1 = u2;
%%%%%%%%%%%%%%%%%%%%%%%%%
%Size of \Omega_1 = s1 =(m/D)+overlaphalf
%\Gamma_1 = s1+1
s=floor(m/D);
Eud(:,:,1)=zeros(m,n);
Eud(1:s-overlaphalf,:,1)=u(1:s-overlaphalf,:);
%On the overlapping part \Omega_1 \cap \Omega_2 we define u1=u/2;
Eud(s-overlaphalf+1:s+overlaphalf,:,1)=u(s-overlaphalf+1:s+overlaphalf,:)/2;
%Correction function for u_1
Chi(:,:,1)=zeros(m,n);
Chi(1:s-overlaphalf,:,1)=1;
for j=s-overlaphalf+1:s+overlaphalf
Chi(j,:,1)=1-1/overlap*(j-(s+overlap/2-overlap+1));
```

```
end
Chi(s+overlaphalf+1:m,:,1)=0;
if D>2
for i=2:D-1
Eud(:,:,i)=zeros(m,n);
Eud((i-1)*s-overlaphalf+1:(i-1)*s+overlaphalf,:,i)=u((i-1)*s-overlaphalf+1:(i-1)*s+overlaphalf,:)/2;
Eud((i-1)*s+overlaphalf+1:i*s-overlaphalf,:,i)=u((i-1)*s+overlaphalf+1:i*s-overlaphalf,:);
Eud(i*s-overlaphalf+1:i*s+overlaphalf,:,i)=u(i*s-overlaphalf+1:i*s+overlaphalf,:)/2;
%Correction function for u_i
Chi(:,:,i)=zeros(m,n);
for j=(i-1)*s-overlaphalf+1:(i-1)*s+overlaphalf
Chi(j,:,i)=1/overlap*(j-((i-1)*s+overlaphalf-overlap+1));
end
Chi((i-1)*s+overlaphalf+1:i*s-overlaphalf,:,i)=1;
for j=(i)*s-overlaphalf+1:i*s+overlaphalf
Chi(j,:,i)=1-1/overlap*(j-(i*s+overlaphalf-overlap+1));
end
end
end
%Size of \Omega_D = m-(D-1)*s+overlaphalf
%\Gamma_2 = (D-1)*s-overlaphalf
Eud(:,:,D)=zeros(m,n);
Eud((D-1)*s-overlaphalf+1:(D-1)*s+overlaphalf,:,D)=u((D-1)*s-overlaphalf+1:(D-1)*s+overlaphalf,:)
Eud((D-1)*s+overlaphalf+1:m,:,D)=u((D-1)*s+overlaphalf+1:m,:);
%Correction function for u_2 (BUPU)
Chi(:,:,D)=zeros(m,n);
Chi(1:(D-1)*s-overlaphalf,:,D)=0;
for j=(D-1)*s-overlaphalf+1:(D-1)*s+overlaphalf
Chi(j,:,D)=1/overlap*(j-((D-1)*s+overlaphalf-overlap+1));
end
```

```
Chi((D-1)*s+overlaphalf+1:m,:,D)=1;
%Test if the sum of the Chi's is 1
for i=1:m
if(sum(Chi(i,:,1:D),3)~=1)
fprintf('Chi is not correct choosen')
pause()
end
end
%%%%%%%%%%%%%
uplot(:,:,1)=zeros(m,n); uplot(:,:,2)=zeros(m,n); uplot(:,:,3)=zeros(m,n);
uplot(:,:,1)=u; uplot(:,:,2)=u; uplot(:,:,3)=u;
for i=1:D-1
uplot(i*s-overlaphalf,:,1)=1;
uplot(i*s-overlaphalf,:,2:3)=0;
uplot(i*s+overlaphalf+1,:,1:2)=0;
uplot(i*s+overlaphalf+1,:,3)=1;
end
figure
imagesc(uplot,clims); axis image; axis off; title('Initial Picture');
pause(0.01)
%%%%%%%%%%%%%%%%%%%
%
%rect = get(gcf,'Position');
%rect(1:2) = [0 0];
%%%%%% MOVIE %%%%%%%%%%
figure
%count number of necessary outer iterations:
r=0;
err=1;
while err > errthresh
```

```
r=r+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%
%% ITERATION FOR U1 IN \OMEGA_1 %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
%Computation only takes place on \Omega_1 + \Gamma_1
for k=1:sub
g1=lambda(1:s+overlaphalf+etacomp2+1,:).*(g(1:s+overlaphalf+etacomp2+1,:)-...
sum(Eud(1:s+overlaphalf+etacomp2+1,:,2:D),3));
y1=sum(Eud(1:s+overlaphalf+etacomp2+1,:,2:D),3);
z1=Eud(1:s+overlaphalf+etacomp2+1,:,1)+...
lambda(1:s+overlaphalf+etacomp2+1,:).*(g1-Eud(1:s+overlaphalf+etacomp2+1,:,1));
%FIXED POINT ITERATION FOR \ETA only within a small stripe around \Gamma_1:
eta1=zeros(s+overlaphalf+etacomp2+1,n);
etatr=eta1(s+overlaphalf-etacomp1+1:s+overlaphalf+etacomp2+1,:);
ytr=y1(s+overlaphalf-etacomp1+1:s+overlaphalf+etacomp2+1,:);
ztr=z1(s+overlaphalf-etacomp1+1:s+overlaphalf+etacomp2+1,:);
d=1;
it=0;
while (d > erreta*sum(sum(abs(etatr(:,:,1))))/100) & (it<itetamax)
ftr=etatr(:,:,1)-ytr-ztr;
%Chambolle
divp=proj(ftr,lambda0,dt,projerr);
etatr1=ztr+lambda0.*divp;
etatr1(1:etacomp1,1:n)=0;
if etacomp2>0
etatr1(etacomp1+2:etacomp1+etacomp2+1,1:n)=0;
end
d=sum(sum(abs(etatr(:,:,1)-etatr1)));
etatr(:,:,1) = etatr1;
clear etatr1
```

```
    it=it+1;
    end
    clear f divp it
    %We compute the projection for u_1 only in Omega_1
    eta1(s+overlaphalf+1,:)=etatr(etacomp1+1,:);
    f1=y1+z1-eta1;
    divp1=proj(f1,lambda0,dt,projerr);
    clear f y_omega1
    %%%%%%%%%% SOLUTION E_1u_1^{(n+1)} of minimization problem in \OMEGA_1:
    Eud(1:s+overlaphalf+etacomp2+1,:,1)=z1-eta1-lambda0.*divp1;
    Eud(s+overlaphalf+1:m,:,1) = 0;
    clear z_omega1 divp_omega1 eta_omega1
    end
    u1new=sum(Eud(:,:,1:D),3);
    uplot(:,:,1)=zeros(m,n); uplot(:,:,2)=zeros(m,n); uplot(:,:,3)=zeros(m,n);
    uplot(:,:,1)=u1new; uplot(:,:,2)=u1new; uplot(:,:,3)=u1new;
    for i=1
    uplot(i*s+overlaphalf+1,:,1:2)=0;
    uplot(i*s+overlaphalf+1,:,3)=1;
    end
    imagesc(uplot,clims); axis image; axis off; colormap(gray);title([num2str(r) ' iterations
']);
    pause(0.01)
    %%%%%%%%%% %%% %%%%%%%%
    mov_matrix(D*(r-1)+1) = getframe(gcf,rect);
    %%%%%%%%%% %% %%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%
    %% ITERATIONs FOR Ui IN \OMEGA_i FOR 1<i<D%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for i=2:D-1
```

```
for k=1:sub
gd=lambda((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:).*...
(g((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:)-...
sum(Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,1:i-1),3)-...
sum(Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,i+1:D),3));
y=sum(Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,1:i-1),3)+...
sum(Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,i+1:D),3);
z=Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,i)+...
lambda((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:).*...
(gd-Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,i));
%FIXED POINT ITERATION FOR \ETA only within a small stripe around \Gamma_i:
%Compute fixed point iteration for \eta only in a small stripe around
%\Gamma_i\cap\Omega_{i-1}
etai=zeros(s+2*overlaphalf+etacomp2+2,n);
etatr_L=etai(1:etacomp1+etacomp2+1,:);
ytr_L=y(1:etacomp1+etacomp2+1,:);
ztr_L=z(1:etacomp1+etacomp2+1,:);
d=1;
it=0;
while (d > erreta*sum(sum(abs(etatr)))/100) & (it<itetamax)
ftr=etatr_L-ytr_L-ztr_L;
divp=proj(ftr,lambda0,dt,projerr);
etatr1_L=ztr_L+lambda0.*divp;
if etacomp2>0
etatr1_L(1:etacomp2,:)=0;
end
etatr1_L(etacomp2+2:etacomp2+etacomp1+1,1:n)=0;
d=sum(sum(abs(etatr_L-etatr1_L)));
etatr_L = etatr1_L;
clear etatr1_L
```

```
it=it+1;
end
clear ftr divp it y_stripeL z_stripeL
%Compute fixed point iteration for \eta only in a small stripe around
%\Gamma_i\cap\Omega_{i+1}
etatr_R=etai(s+2*overlaphalf-etacomp1+2:s+2*overlaphalf+etacomp2+2,:);
ytr_R=y(s+2*overlaphalf-etacomp1+2:s+2*overlaphalf+etacomp2+2,:);
ztr_R=z(s+2*overlaphalf-etacomp1+2:s+2*overlaphalf+etacomp2+2,:);
d=1;
it=0;
while (d > erreta*sum(sum(abs(etatr_R)))/100) & (it<itetamax)
ftr=etatr_R-ytr_R-ztr_R;
divp=proj(ftr,lambda0,dt,projerr);
etatr1_R=ztr_R+lambda0.*divp;
etatr1_R(1:etacomp1,1:n)=0;
if etacomp2>0
etatr1_R(etacomp1+2:etacomp1+etacomp2+1,1:n)=0;
end
d=sum(sum(abs(etatr_R-etatr1_R)));
etatr_R = etatr1_R;
clear eta1_stripeR
it=it+1;
end
clear ftr divp it ytr_R ztr_R
%We compute the projection for u_i only in Omega_i
etai(etacomp2+1,:)=etatr_L(etacomp2+1,:);
etai(s+2*overlaphalf+1,:)=etatr_R(etacomp1+1);
f=y+z-etai;
divpi=proj(f,lambda0,dt,projerr);
clear f y_omegad
```

```
%%%%%%%%%%%% SOLUTION Eu_i^{(n+1)} of minimization problem in \OMEGA_i:
Eud((i-1)*s-overlaphalf-etacomp2:i*s+overlaphalf+etacomp2+1,:,i)=z-etai-lambda0.*divpi;
Eud(1:(i-1)*s-overlaphalf-1,:,i) = 0;
Eud(i*s+overlaphalf+1:m,:,i) = 0;
clear z_omegad divp_omegad eta_omegad
%We compute the projection for u_D only in Omega_D
%etaD(etacomp2+1,:)=etatr(etacomp2+1,:);
%fD=yD+zD-etaD;
%divpD=proj_convset(fD,lambda0,dt,projerr);
%clear fD yD
%
%%%%%%%%%%%% SOLUTION Eu_D^{(n+1)} of minimization problem in \OMEGA_D:
%Eud((D-1)*s-overlaphalf-etacomp2:m,:,D)=zD-etaD-lambda0.*divpD;
%Eud(1:(D-1)*s-overlaphalf-1,:,D) = 0;
%clear zD divpD etaD
end
uinew=sum(Eud(:,:,1:D),3);
uplot(:,:,1)=zeros(m,n); uplot(:,:,2)=zeros(m,n); uplot(:,:,3)=zeros(m,n);
uplot(:,:,1)=uinew; uplot(:,:,2)=uinew; uplot(:,:,3)=uinew;
uplot(i*s+overlaphalf+1,:,1:2)=0;
uplot(i*s+overlaphalf+1,:,3)=1;
imagesc(uplot,clims); axis image; axis off; colormap(gray);title([num2str(r) ' iterations
']);
pause(0.01)
%%%%%%%%%%% MOVIE %%%%%%%%%%%%%%%%%
mov_matrix(D*(r-1)+i) = getframe(gcf,rect);
%%%%%%%%%%% MOVIE %%%%%%%%%%%%%%%%%
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% ITERATION FOR UD IN \OMEGA_D %%%%%%%%%%%
```

```
for k=1:sub
gD=lambda((D-1)*s-overlaphalf-etacomp2:m,:).*(g((D-1)*s-overlaphalf-etacomp2:m,:)-...
sum(Eud((D-1)*s-overlaphalf-etacomp2:m,:,1:D-1),3));
yD=sum(Eud((D-1)*s-overlaphalf-etacomp2:m,:,1:D-1),3);
zD=Eud((D-1)*s-overlaphalf-etacomp2:m,:,D)+...
lambda((D-1)*s-overlaphalf-etacomp2:m,:).*(gD-Eud((D-1)*s-overlaphalf-etacomp2:m,:,D));
%FIXED POINT ITERATION FOR \ETA only within a small stripe around \Gamma_D:
%Compute fixed point iteration for \eta
etaD=zeros(m-(D-1)*s+overlaphalf+etacomp2+1,n);
etatr=etaD(1:etacomp1+etacomp2+1,:);
ytr=yD(1:etacomp1+etacomp2+1,:);
ztr=zD(1:etacomp1+etacomp2+1,:);
d=1;
it=0;
while (d > erreta*sum(sum(abs(etatr)))/100) & (it<itetamax)
ftr=etatr-ytr-ztr;
divp=proj(ftr,lambda0,dt,projerr);
etatr1=ztr+lambda0.*divp;
if etacomp2>0
etatr1(1:etacomp2,:)=0;
end
etatr1(etacomp2+2:etacomp2+etacomp1+1,1:n)=0;
d=sum(sum(abs(etatr-etatr1)));
etatr = etatr1;
clear etatr1
it=it+1;
end
clear f divp it
%We compute the projection for u_D only in Omega_D
etaD(etacomp2+1,:)=etatr(etacomp2+1,:);
```

```
fD=yD+zD-etaD;
divpD=proj_convset(fD,lambda0,dt,projerr);
clear fD yD
%%%%%%%%%%% SOLUTION Eu_D^{(n+1)} of minimization problem in \OMEGA_D:
Eud((D-1)*s-overlaphalf-etacomp2:m,:,D)=zD-etaD-lambda0.*divpD;
Eud(1:(D-1)*s-overlaphalf-1,:,D) = 0;
clear zD divpD etaD
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%SOLUTION ON WHOLE DOMAIN:%%%%%%%
unew=sum(Eud(:,:,1:D),3);
err = sum(sum(abs(u-unew))./(m*n)
u=unew;
clear unew
for i=1:D
Eud(:,:,i)=Chi(:,:,i).*u;
end
imagesc(u,clims); axis image; axis off; colormap(gray);title([num2str(r) ' iterations ']);%,
num2str(cputime-t) 'sec']);
pause(0.01)
%%%%%%%%%%% MOVIE %%%%%%%%%%%
mov_matrix(D*(r-1)+D) = getframe(gcf,rect);
%%%%%%%%%%% MOVIE %%%%%%%%%%%
end
iterations=r
fprintf('Time: %f sec \n', cputime-t);
%%%%%%%MOVIE%%%%%%%%%%%%%%%%%%
movie(gcf,mov_matrix,1,50);
movie2avi(mov_matrix,[savepath 'movie.avi'],'compression','Indeo5','fps',2);
%%%%%%%%MOVIE%%%%%%%%%%%%%%%%%
```

```
%PROJECTION - CHAMBOLLE:
%Iteration to compute the projection
%\Pi_{\lambda K})(a+A^*(g-Aa)) (Chambolle)
function divp=proj(f,lambda0,dt,projerr)
[m,n]=size(f);
px(1:m,1:n)=0;
py(1:m,1:n)=0;
p=sparse([px,py]);
diff=1;
it=0;
while diff > projerr * sum(sum(abs(p)))/(n*m)
it=it+1;
%divergence (backward differences)
divp(2:m-1,2:n-1) = (px(2:m-1,2:n-1)-px(1:m-2,2:n-1)) + (py(2:m-1,2:n-1)-py(2:m-1,1:n-
2));
divp(2:m-1,1) = (px(2:m-1,1)-px(1:m-2,1)) + py(2:m-1,1);
divp(2:m-1,n) = (px(2:m-1,n)-px(1:m-2,n)) - py(2:m-1,n-1);
divp(1,2:n-1) = px(1,2:n-1) + (py(1,2:n-1)-py(1,1:n-2));
divp(1,1) = px(1,1) + py(1,1);
divp(1,n) = px(1,n) - py(1,n-1);
divp(m,2:n-1) = -px(m-1,2:n-1) + (py(m,2:n-1)-py(m,1:n-2));
divp(m,1) = -px(m-1,1) + py(m,1);
divp(1,n) = -px(m-1,n) - py(m,n-1);
arg = divp - f./lambda0;
%gradient (forward differences)
gradx(1:m-1,1:n) = arg(2:m,1:n)-arg(1:m-1,1:n);
gradx(m,1:n) = 0;
grady(1:m,1:n-1) = arg(1:m,2:n)-arg(1:m,1:n-1);
grady(1:m,n) = 0;
grad = [gradx, grady];
```

```
if sum(sum(grad==0))==n*m
absgrad=grad;
else
absgrad = abs(grad);
end
p1 = (p + dt.*grad)./(1+dt.*absgrad);
diff = sum(sum(abs(p-p1)))/(2*m*n);
p=p1;
clear p1;
px(1:m,1:n) = p(1:m,1:n);
py(1:m,1:n) = p(1:m,n+1:2*n);
if it>100
it
end
end
```

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. Theory52(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. Theory52(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 Applications4(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 Controle et Equations aux Dérivées Partielles4, 1998, pp. 83–96

[23] A. Chambolle, An algorithm for total variation minimization and applications, J. Math. Imaging Vision20(1-2), 2004, pp. 89–97.

[24] T.F. Chan, G.H.Golub, andP. Mulet, A nonlinear primal-dual method for total variation-based image restoration, SIAM J. Sci. Comput.20(6), 1999, pp. 1964–1977.

[25] T. F. Chan, and T. P. Mathew, Domain decomposition algorithms, Acta Numerica3, 1994, pp. 61–143.

[26] P. G. Ciarlet, The Finite Element Methods for Elliptic Problems, Classics Appl. Math. 40, SIAM, Philadelphia, 2002

[27] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore, Harmonic Analysis of the space BV, Rev. Mat. Iberoamericana19, 2003, pp. 235–263.

[28] 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.

[29] P. L. Combettes and V. R. Wajs, Signal recovery by proximal for ward backward splitting, Multiscale Model. Simul.4(4), 2005, pp. 1168–1200.

[30] G. David, Global Minimizers of the Mumford-Shah Functional, Current developments in mathematics, 1997 (Cambridge, MA), Int. Press, Boston, MA, 1999, pp. 219–224.

[31] I.Daubechies,M.Defrise,and C.DeMol, An iterative thresholding algorithm for linear inverse problems, Comm. PureAppl.Math.57(11),2004,pp.1413–1457.

[32] J. Darbon and M. Sigelle, A fast and exact algorithm for total variation minimization, IbPRIA 20053522(1), 2005, pp. 351-359.

[33] 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.

[34] G. Dal Maso, An Introduction to Γ-Convergence, Birkhäuser, Boston, 1993.

[35] E. De Giorgi, Convergence problems for functionals and operators, Recent methods in nonlinear analysis, Proc. Int. Meet., Rome 1978, 1979, pp. 131188.

[36] 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.

[37] I. Ekeland and R. Temam, Convex analysis and variational problems. Translated by Minerva Translations, Ltd., London.,StudiesinMathematicsandits Applications. Vol. 1. Amsterdam - Oxford: North-Holland Publishing Company; New York: American Elsevier Publishing Company, Inc., 1976.

[38] 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

[39] M.Fornasier, Domain decomposition methods for linear inverse problems with sparsity constraints, Inverse Problems23, 2007, pp. 2505–2526.

[40] M. Fornasier, A. Langer, and C.-B. Schönlieb, A convergent overlapping domain decomposition method for total variation minimization, Numerische Mathematik116(4), 2010, pp. 645–685.

[41] 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.

[42] M.Fornasier and C.-B.Schönlieb, Subspace correction methods for total variation and ℓ^1-minimization, SIAMJ.Numer.Anal.47(5),2009,pp.3397–3428

[43] M. Fornasier and R. Ward, Iterative thresholding meets free-discontinuity problems, Found. Comput. Math.10(5), 2010, pp. 527–567.

[44] T.Goldstein,S.Osher,The split Bregman method for L^1 regularized problems, UCLA CAM Report 08-29, 2008.

[45] T. Kohlberger, C. Schnorr, A. Bruhn, and J. Weickert. Domain decomposition for variational optical-flow computation. IEEE Transactions on Image Processing, 14(8):1125–1137, 2005.

[46] M.L. Hadji, M. Maouni and F.Z. Nouri. *A Wavelet Inpainting by a Toxitrop Model.* iv, 2010 14th International Conference Information Visualisation, 2010, IEEE ISBN: 978-0-7695-4165-5, doi.ieee computer society.org/10.1109/IV.2010.96, pp. 559-563.

[47] A. Langer , PhD-thesis "Subspace Correction and Domain Decomposition Methods for Total Variation Minimization" UNIVERSITÄT LINZ JOHANNES KEPLER Juliy 2011.

[48] A. Langer, S. Osher, and M. Fornasier, Bregmanized Domain Decomposition for Image Restoration, submitted to Journal of Scientific Computing, pp. 25.

[49] P.-L. Lions, On the Schwarz alternating method I. In R. Glowinski, G.H. Golub, G.A. Meurant, J. P´eriaux (eds.), First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1988, pp. 1–42.

[50] D.Mumford and J.Shah, Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems, Comm. Pure Appl. Math.42, 1989, pp. 577–685.

[51] Y. Nesterov, A method for unconstrained convex minimization problem with the rate of convergence $O(\frac{1}{k^2})$, Doklady ANSSSR (translated as Soviet Math. Docl.)269, 1983, pp. 543–547.

[52] 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.

[53] 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.

[54] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, An algorithm for Minimizing the Mumford-Shah Functional,In ICCV Proceedings, LNCS,Springer, 2009

[55] 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.

[56] L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D60(1-4), 1992, pp. 259–268.

[57] H.A. Schwarz, Gesammelte mathematische abhandlungen, Vierteljahrsschrift der Naturforschenden Gesellschaft 15 (1870) 272–286.

[58] X.-C. Tai and J. Xu, Global convergence of subspace correction methods for convex optimization problems, Math. Comp.71(237), 2002, pp. 105–124.

[59] A. Toselli and O.Widlund, Domain Decomposition Methods - Algorithms and Theory, Springer Series in Computational Mathematics, Springer, Berlin, 2005

[60] Tikhonov, A. N.; V. Y. Arsenin (1977). Solution of Ill-posed Problems. Washington: Winston & Sons. ISBN 0-470-99124-0.

[61] L. Vese, A study in the BV space of a denoising-deblurring variational problem., Appl. Math. Optim.44, 2001, pp. 131–161.

[62] J. von Neumann, The geometry of orthogonal spaces, Functional operators- vol. II, Annals of Math. Studies, no. 22, Princeton University Press, 1950, This is a reprint of mimeographed lecture notes, first distributed in 1933. MR 11:599e

[63] 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.

[64] J. Xu, The method of subspace corrections, J. Comp. Appl. Math. 128, 2001, pp. 335–362

[65] 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.

[66] 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.

[67] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. SIAM Journal on Scientific Computing, 17(1):227–238, 1996.

[68] C.R. Vogel, M.E. Oman, et al. Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Transactions on Image Processing, 7(6):813–824, 1998.

[69] Y.Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. SIAM Journal on Imaging Sciences, 1(3):248–272, 2008.

[70] Y. Wang, W. Yin, and Y. Zhang. A fast fixed-point algorithm for convex total variation regularization. Technical report, working paper, Rice University, Houston, TX, 2007.

[71] Y. Wang, W. Yin, and Y. Zhang. A fast algorithm for image deblurring with total variation regularization. Rice University CAAM Technical Report TR07-10, 2007.

[72] J.Weickert. Recursive separables chemes for nonlinear diffusion filters. Lecture Notes in Computer Science, 1252:260–271, 1997.

[73] J. Weickert, B. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Transactions on Image Processing, 7(3):398–410, 1998.

[74] Y.W. Wen, M.K. Ng, and Y.M. Huang. Efficient total variation minimization methods for color image restoration. IEEE Transactions on Image Processing, 17(11):2081–2088, 2008.

[75] J.C. Xu. Iterative methods by space decomposition and subspace correction. SIAM Rev., 34(4):581–613, 1992.

[76] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for compressend sensing and related problems. SIAM J. Imaging Sciences, 1(1):143–168, 2008.

[77] F.Z. Zeghbib, M.Maouni & F.Z. Nouri, Overlapping and non overlapping domain decomposition methods for image restoration .Int. J.Appl. Math. Stat. ; Vol 40 ; Issue No. 10, 2013 ; p.123-128. www.ceserp.co/Appl.Maths.