SPLITTING METHODS IN IMAGE PROCESSING

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von

Dipl.-Math. Simon Setzer
aus Heidelberg

Mannheim, September 2009

Dekan:	Professor Dr. Felix Freiling, Universität Mannheim
Referent:	Professor Dr. Gabriele Steidl, Universität Mannheim
Korreferent:	Professor Dr. Christoph Schnörr, Universität Heidelberg

Tag der mündlichen Prüfung: 9. Dezember 2009

Acknowledgments

Foremost, I would like to express my sincere gratitude to my thesis supervisor Gabriele Steidl for here guidance, support and constant encouragement. Many thanks go to Tanja Teuber whom I had the great pleasure to work with in the last two years. I also thank all my other collaborators: I am indebted to Raymond Chan (Chinese University of Hong Kong) for his fruitful collaboration and his invitation to the workshop in Singapore last year. My thanks also go to Bernhard Burgeth (Saarland University), Stephan Didas (Saarland University, now Fraunhofer-Institut, Kaiserslautern), Guido Morkoette (University of Mannheim) and Björn Popilka for the constructive and inspiring collaboration.

Finally, I want to express my deep gratitude to my family, to the Gerner family and especially to Yela for everything they have have done for me.

Summary

It is often necessary to restore digital images which are affected by noise (denoising), blur (deblurring), or missing data (inpainting). We focus here on variational methods, i.e., the restored image is the minimizer of an energy functional.

The first part of this thesis deals with the algorithmic framework of how to compute such a minimizer. It turns out that operator splitting methods are very useful in image processing to derive fast algorithms. The idea is that, in general, the functional we want to minimize has an additive structure and we treat its summands separately in each iteration of the algorithm which yields subproblems that are easier to solve. In our applications, these are typically projections onto simple sets, fast shrinkage operations, and linear systems of equations with a nice structure.

The two operator splitting methods we focus on here are the forward-backward splitting algorithm and the Douglas-Rachford splitting algorithm. We show based on older results that the recently proposed alternating split Bregman algorithm is equivalent to the Douglas-Rachford splitting method applied to the dual problem, or, equivalently, to the alternating direction method of multipliers. Moreover, it is illustrated how this algorithm allows us to decouple functionals which are sums of more than two terms.

In the second part, we apply the above techniques to existing and new image restoration models. For the Rudin-Osher-Fatemi model, which is well suited to remove Gaussian noise, the following topics are considered: we avoid the staircasing effect by using an additional gradient fitting term or by combining first- and second-order derivatives via an infimal-convolution functional. For a special setting based on Parseval frames, a strong connection between the forward-backward splitting algorithm, the alternating split Bregman method and iterated frame shrinkage is shown. Furthermore, the good performance of the alternating split Bregman algorithm compared to the popular multistep methods is illustrated. A special emphasis lies here on the choice of the step-length parameter. Turning to a corresponding model for removing Poisson noise, we show the advantages of the alternating split Bregman algorithm in the decoupling of more complicated functionals. For the inpainting problem, we improve an existing wavelet-based method by incorporating anisotropic regularization techniques to better restore boundaries in an image. The resulting algorithm is characterized as a forward-backward splitting method. Finally, we consider the denoising of a more general form of images, namely, tensor-valued images where a matrix is assigned to each pixel. This type of data arises in many important applications such as diffusion-tensor MRI.

Zusammenfassung

In vielen Anwendungen der Bildverarbeitung ist es notwendig, Rauschen und Blur aus digitalen Bildern zu entfernen (Entrauschen und Deblurren), sowie unbekannte Regionen in Bildern wiederherzustellen (Inpainting). Wir betrachten hier sogenannte Variationsmethoden, d.h. das restaurierte Bild ist ein Minimierer eines Energiefunktionals.

Im ersten Teil dieser Arbeit werden Algorithmen zur Berechnung eines solchen Minimierers betrachtet. Dabei stellt sich heraus, dass sogenannte Operator-Splitting-Verfahren in der Bildverarbeitung besonders nützlich sind, denn das zu minimierende Funktional hat im Allgemeinen eine additive Struktur. Diese ermöglicht es, die Summanden in jeder Iteration separat zu betrachten, was zu einfacheren Teilproblem führt. In unseren Anwendungen sind das zum Beispiel Projektionen auf einfache Mengen, schnelle Shrinkage-Operationen und das Lösen linearer Gleichungssysteme von einfacher Struktur. Zwei Operator-Splitting-Methoden sind in dieser Arbeit von besonderem Interesse, nämlich der Forward-Backward-Splitting-Algorithmus und der Douglas-Rachford-Splitting-Algorithmus. Wir zeigen, basierend auf älteren Resultaten, dass der kürzlich vorgestelle Alternating-Split-Bregman-Algorithmus äquivalent zum Douglas-Rachford-Splitting-Algorithmus für das duale Problem und zur Alternating direction method of multipliers ist. Desweiteren wird untersucht, wie dieses Verfahren verwendet werden kann, um Zielfunktionen mit mehr als zwei Summanden zu entkoppeln.

Im zweiten Teil dieser Arbeit wenden wir die obigen Verfahren auf bestehende und neue Modelle zur Bildrestauration an. Zunächst betrachten wir das Rudin-Osher-Fatemi-Model zum Entrauschen von Bildern unter der Annahme von Gaußschem Rauschen: Zur Vermeidung des sogannten Staircasing-Effekts verwenden wir einen zusätzlichen Ähnlichkeitsterm bezüglich des Gradienten oder einen Infimal-Convolution-Term mit ersten und zweiten Ableitungen. Für einen Spezialfall basierend auf Parseval-Frames beleuchten wir die enge Verbindung zwischen dem Forward-Backward-Splitting-Algorithmus, dem Alternating-Split-Bregman-Algorithmus und dem iteriertem Frame-Shrinkage. Außerdem zeigen wir die gute Leistung des Alternating-Split-Bregman-Algorithmus im Vergleich zu populären Multistep-Methoden. Hierbei untersuchen wir insbesondere den Einfluss des Schrittweitenparameters. Die Vorteile des Alternating-Split-Bregman-Algorithmus werden besonders deutlich, wenn wir ein verwandtes aber schwieriger zu minimierendes Model zum Entfernen von Poisson-Rauschen betrachten. Zum Lösen des Inpainting-Problems erweitern wir einen Wavelet-basierten Ansatz durch Techniken der anisotropen Regularisierung. Dies trägt dazu bei, dass Kanten im Bild besser wiederhergestellt werden. Auf diese Weise erhalten

wieder einen Forward-Backward-Splitting-Algorithmus. Abschließend behandeln wir das Entrauschen einer allgemeineren Form von digitalen Bildern, nämlich tensorwertige Bilder. Dabei ist jedem Pixel eine Matrix zugeordnet. Diese Daten treten in vielen wichtigen Anwendungen auf, z.B. in der Diffusionstensor-MRT.

Contents

1 Introduction **1**

2 Variational methods **7**
2.1 Introduction 7
2.2 Optimization problems, duality and proximation 7
2.3 Operator splittings 13
2.3.1 Weak convergence of Picard iterations of averaged operators 13
2.3.2 Forward–backward splitting 23
2.3.3 Douglas–Rachford splitting 24
2.3.4 Other operator splittings 26
2.4 Bregman methods 28
2.4.1 Bregman proximal point method 28
2.4.2 Split Bregman and augmented Lagrangian method 34
2.4.3 Alternating split Bregman algorithm 40
2.4.4 Multiple splittings 45

3 Application to image denoising **51**
3.1 Introduction 51
3.2 Continuous denoising models 52
3.3 Discrete models for Gaussian noise removal 56
3.4 DRS and FBS for Gaussian noise removal 61
3.4.1 DRS algorithm 61
3.4.2 FBS algorithm 62
3.4.3 Relation between DRS and FBS algorithm 63
3.4.4 Geometrical interpretation of DRS and FBS algorithm 66
3.5 Comparison of DRS and FBS with multistep methods 69
3.5.1 Multistep methods 70
3.5.2 Numerical experiments 73
3.6 Some generalizations of the ROF model 78
3.7 Poisson noise removal 83

4 Application to image inpainting **87**
4.1 Introduction . . . 87
4.2 Anisotropic regularization and diffusion . . . 90
4.3 Anisotropic Haar-wavelet shrinkage . . . 92
4.4 Convergence considerations . . . 97
4.5 Numerical examples . . . 101

5 Application to the denoising of matrix fields **111**
5.1 Introduction . . . 111
5.2 Preliminaries . . . 113
5.3 Component-based regularization . . . 115
5.4 Operator-based regularization . . . 117
5.5 Numerical results . . . 120

A Wavelet-frames **127**
A.1 Frames . . . 127
A.2 Framelets . . . 128

B Matrix diagonalization via cosine transform **137**

C Chambolle's semi-implicit gradient descent algorithm **139**

CHAPTER 1

Introduction

Images are a main source of human information about the world. Hence, a lot of research is conducted in medicine, psychology, mathematics and computer science, on the one hand, to understand the human visual system and, on the other hand, to implement image processing tasks on computers. For the first approach we refer, e.g., to [105, 112] and note that the human visual system might even be the key to unlocking the mysteries of the highest brain functions, see, e.g., [137].

We focus in this paper on the second approach, more precisely, on the following classical topics of image restoration: denoising, deblurring and inpainting. In the noise-free case, the latter is a special form of interpolation. Among others, we also deal with current problems of denoising in the presence of non-Gaussian noise and the restoration of tensor-valued data.

To tackle these image processing problems one must answer the following two classes of questions which are clearly intertwined:

i) **Modeling of the image processing problem:** What features in an image are we interested in? What are general assumption on a restored image? Can we specify this probabilistically? Is it possible to model the process which corrupts our data, e.g., the statistical properties of the noise?

ii) **Numerical solution of the problem:** Which methods can be used to compute the restored image according to the chosen model? Are the corresponding problems solvable in a reasonable amount of time?

In this paper, we model our image processing tasks as minimization problems. The restored image is then given as a minimizer of the corresponding problem. Often, the term *variational methods* is used for this approach. Other approaches include for example PDE-based methods. The energy functionals we use in our minimization problems constitute our answers to the class i) of questions above. In our applications, the functional to minimize is the sum of two terms, a *data-fitting term*, which relates our minimizer to the given corrupted data and a so-called *regularization term* which incorporates the method's "view

of the world": It might assign smaller energies to certain features like smooth areas or it might prefer texture or sharp object boundaries.

As for the second set of question ii), we note that the success of many variational methods is related to the fact that the corresponding objective functions contain terms which are convex but not differentiable or they may include constraints. This is the case, e.g., for the Rudin-Osher-Fatemi (ROF) model of [187] and the corresponding dual problem, respectively. Bearing in mind the success of *interior point methods*, cf. [121], one idea might be to use them for our image processing problems. One would then use the primal and the dual problem together, avoid constraints and non-differentiable terms via differentiable barrier functions and find a solution by means of Newton steps.

However, for our examples, we only need a modest accuracy and, more importantly, our minimization problems are structured in ways that encourage us to use algorithms that decompose the problem into easier parts. For example, the constraints we encounter are often very simple sets and the projection onto them is cheap. Hence, *splitting methods* have become very popular to solve variational problems in image processing and they are the main topic of this paper. One of the first such splitting method used in image processing was a simple gradient descent algorithm for the dual problem of the ROF model. The constraint which appears in this model was incorporated by projecting onto the corresponding set after each gradient descent step. This algorithms can be characterized as a special *forward-backward splitting method*. Recently, the *alternating split Bregman algorithm* was proposed by Goldstein and Osher in [120] for the ROF denoising and deblurring problem. This method was originally derived as a special case of a *Bregman proximal point algorithm*. The main idea of the Bregman proximal point algorithms is to transform the problem into easier subproblems by adding a so-called cost-to-move term. However, in the derivation of the alternating split Bregman algorithm a variable splitting idea of [217] was used. This encouraged us to interpret this algorithm also as a splitting method: It is a *Douglas-Rachford splitting* algorithm applied to the dual problem. In this way, we were able to proof convergence of the alternating split Bregman method. Apart from characterizing the alternating split Bregman algorithm as a Bregman method and a splitting method, we can also, as underlined by Esser in [101], take a third point of view and interpret it as a *primal-dual Lagrangian algorithm*, namely, the *alternating direction method of multipliers* of [109, 110, 117] which updates alternatingly the primal and dual variable in a special way.

To summarize, we have three possibilities to derive the alternating split Bregman algorithm: It is an operator splitting, a special Bregman and a primal-dual Lagrangian method.

Alternating Split Bregman Algorithm	=	Douglas-Rachford Splitting Algorithm for the Dual Problem	=	Alternating Direction Method of Multipliers

It should be mentioned here that Combettes deserves credit for drawing attention in recent years to splitting methods like the forward-backward splitting and the Douglas-Rachford splitting method and their applications in image processing.

The primal-dual Lagrangian point of view was recently used to derive the primal-dual hybrid gradient algorithm of [235]. After this thesis was completed, it was also shown that the primal-dual hybrid gradient algorithm, like the alternating split Bregman algorithm, has a relation to well-known methods in optimization theory, cf. [102]. This very recent result indicates that there is still a lot of potential for new efficient image processing algorithms which use splitting, Bregman and primal-dual ideas, especially as models with more complicated objective functions arise. Impulses are likely to come from adaptive step size strategies, see [234] and Subsection 3.5.2, as well as from multistep methods, cf. Section 3.5.

Contributions

Foundations of the alternating split Bregman algorithm. We show based on older results that the alternating split Bregman algorithm is equivalent to applying an operator splitting, namely the Douglas-Rachford splitting, to the dual problem of our given minimization problem. This allows us to conclude convergence of the alternating split Bregman algorithm.

For two special models used for Gaussian noise removal, we can establish a relation of the alternating split Bregman method to the forward-backward splitting algorithm and to wavelet shrinkage. More precisely, we have for a Besov norm model using a Parseval frame discretization and a special step length the following relation:

Alternating Split Bregman Algorithm	=	Forward-Backward Splitting Algorithm	=	Iterated Frame Shrinkage

For the ROF model we can also apply a Parseval frame-based gradient discretization and a special step length to obtain:

Alternating Split Bregman Algorithm	≈	Forward-Backward Splitting Algorithm	=	Iterated Frame Shrinkage

Furthermore, we provide for the ROF denoising problem numerical comparisons of the Douglas-Rachford splitting/alternating split Bregman method with first-order multistep methods. In particular, we investigate the role of the step length parameter in the alternating split Bregman algorithm. The good performance of this algorithm is illustrated by interpreting it as a special Newton method, i.e., a *second-order* method.

Multiple splittings with the alternating split Bregman algorithm. The advantages of the Douglas-Rachford splitting/alternating split Bregman algorithm become especially apparent when we apply it to more complicated functionals. As an example, we show how multiple splittings of functionals for Poissonian noise removal can be used to decompose the original problem into explicitly solvable subproblems.

Higher-order methods in image denoising. There exist various methods to generalize and improve the ROF model. We consider two ways to use both first and *second-order derivatives* which has the advantage that the staircasing effect of ROF denoising is avoided. First, we introduce an additional *gradient fitting* term which does not complicate the resulting minimization problem too much but allows us to include second-oder derivatives. Second, we propose to use first- and second-order derivatives in an *infimal convolution* functional.

Novel wavelet-based inpainting algorithm. Motivated by [51], we develop an inpainting algorithm that performs in each iteration a denoising step which consists in a fast "anisotropic" frame shrinkage and a subsequent step in which we reset the known data. Our contribution is to introduce an *anisotropic regularization* technique into the inpainting model. The idea is that the process which fills in the unknown parts of a given image should be anisotropic in that it detects object boundaries and thus restores them more efficiently. Numerical experiments demonstrate the advantages of our anisotropic algorithm especially at sharp edges.

Operator-based method for the denoising of matrix-valued images. In many application, for example in medical imaging, a matrix is assigned to each pixel instead of just a gray value. We want to transfer successful techniques from the denoising of scalar-valued images to this setting. Clearly, we could simply denoise the gray value images obtained by considering single components of the matrix-valued data. However, this is not a good strategy since, typically, the information is spread over all components. We propose a new model which we call the *operator-based method*. The main idea is that it couples the matrix channels by viewing the matrices as operators which can be concatenated with other useful nonlinear operators. Furthermore, we transfer the infimal convolution approach to the matrix-valued setting in order to improve existing methods by using both first- and second-order derivatives.

Outline

Chapter 2 includes the theoretical foundations of the algorithms used in this paper. We examine in Section 2.3 the convergence properties of iterations of so-called averaged operators. This is then applied to prove the convergence of operator splitting methods, in particular, the forward-backward splitting algorithm and the Douglas-Rachford splitting algorithm. Another way to split a minimization problem into easier parts is via Bregman methods which are studied in Section 2.4. First, we present the Bregman proximal point algorithm which adds an appropriate cost-to-move term to the given objective function and is thus a generalization of the classical proximal point method. Then, a special application of this Bregman method, the split Bregman algorithm, is considered. Based on this we can define the alternating split Bregman algorithm in Subsection 2.4.3 which we characterize

as a Douglas-Rachford splitting method applied to the dual problem. The alternating split Bregman algorithm allows multiple splittings as shown in Subsection 2.4.4.

The first application of the methods of Chapter 2 is to image denoising in Chapter 3. After motivating continuous denoising models with Besov norm and total variation regularizers for both Gaussian and Poisson noise in Section 3.2, we study discrete Gaussian noise removal in more detail in Section 3.3. In Section 3.4, we apply the forward-backward splitting and the Douglas-Rachford splitting/alternating split Bregman method to the ROF model and give a geometrical interpretation of both algorithms. Moreover, the relation between these methods for a special Parseval frame setting is explored which also gives insight into the connection to wavelet shrinkage. In Section 3.5, we compare the forward-backward splitting algorithm and the alternating split Bregman algorithm numerically to two multistep methods, namely Nesterov's algorithm and FISTA. We also draw attention to the role of the step length parameter in the alternating split Bregman algorithm. In Section 3.6, we briefly show two extensions of the ROF method: a gradient-fitting scheme and an infimal convolution regularization term. Finally, we turn to Poisson noise removal in Section 3.7. We show that here the multiple decoupling obtained by using the Douglas-Rachford splitting/alternating split Bregman algorithm is especially useful.

In Chapter 4, we present novel wavelet-based inpainting algorithms. As we explain in Section 4.2 and Section 4.3, these algorithms make use of anisotropic regularization and diffusion techniques. We can guarantee convergence by characterizing the proposed inpainting methods as forward-backward splitting algorithms, cf., Section 4.4. Numerical results can be found in Section 4.5.

Finally, we present an application of our methods to the denoising of tensor-valued images in Chapter 5. This kind of data is important, e.g., in medical imaging, see Section 5.1. We consider a component-based approach in Section 5.3 which we generalize via an infimal convolution regularizer. The advantage of our new operator-based approach, presented in Section 5.4, is that it makes use of the operator structure of matrices. We give numerical examples in Section 5.5.

CHAPTER 2

Variational methods

2.1 Introduction

Many image processing problems can be solved via *variational methods* and the restored image is a minimizer of an appropriate energy functional. After introducing in Section 2.2 the general type of energy functionals we want to deal with in this paper, we consider two main classes of methods to find a minimizer:

First, we consider operator splitting algorithms in Section 2.3 which decompose our original problem into subproblems which are easier to solve. This is possible since we assume that the functionals to minimize have an additive structure. The resulting algorithms are in our case iterations of averaged operators which allows us to conclude convergence. There exist many operator splitting techniques. A special focus lies here on the forward-backward splitting algorithm (Subsection 2.3.2) and the Douglas-Rachford splitting algorithm (Subsection 2.3.3).

Second, we consider Bregman methods in Section 2.4. We recall the basic Bregman proximal point algorithm, a generalization of the classical proximal point algorithm, which yields easier subproblems by adding a term that penalizes the distance to the preceding iterate, see Subsection 2.4.1. Applying the Bregman proximal point algorithm in a special way yields the split Bregman algorithm (Section 2.4.2) and finally the alternating split Bregman method (Section 2.4.3). Both split Bregman methods turn out to be equivalent to classical algorithms, in particular, we show that the latter coincides with a Douglas-Rachford splitting algorithm. In Subsection 2.4.4, we illustrate the advantages of using the alternating split Bregman method to decouple complicated functionals which can be the sum of more than two terms.

2.2 Optimization problems, duality and proximation

As stated in the introduction we are interested in the minimization of nonsmooth functionals. We start by considering basic notations and a simple operator, the proximity

operator, which will prove to be useful in many places. Let H be an infinite-dimensional real Hilbert space. In this paper we try to keep our analysis as long as possible in the infinite-dimensional setting. It should be stressed though that the straightforward way of discretizing before performing the minimization process yields satisfactory results for our examples. So, for all our practical examples we will work in $\mathbb{R}^n$. In this paper we will use the notation $\overline{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$ for the extended reals. Let $\Phi : H \to \overline{\mathbb{R}}$ be a function with the following properties

- Φ is proper, i.e., it is not $+\infty$ everywhere,
- Φ is convex,
- Φ is *lower semi-continuous* (l.s.c.) at each point $u \in H$, i.e., $\Phi(u) \leq \lim_{v \to u} \inf \Phi(v) = \sup_{\epsilon > 0} \inf_{v \in B(u,\epsilon)} \Phi(v)$.

A functional which is l.s.c. on the whole domain is also called *closed* since its epigraph is closed. For a function Φ satisfying the above three properties it is often useful to consider for some parameter $\eta > 0$ the *Moreau envelope* ${}^{\eta}\Phi : H \to \mathbb{R}$, see [160]. It is defined as follows:

$$ {}^{\eta}\Phi(u) := \min_{v \in H} \big\{ \frac{1}{2} \|v - u\|_2^2 + \eta \Phi(v) \big\}. \tag{2.1} $$

Clearly, the objective function in (2.1) is coercive and strictly convex and thus for every $u \in H$, there exists a unique minimizer, compare [97, Proposition 1.2]. This minimizer is called the *proximum* and the mapping which assigns to each $u \in H$ the proximum is called the *proximity operator* $\operatorname{prox}_{\eta\Phi}$ of $\eta\Phi$, i.e., we have

$$ \operatorname{prox}_{\eta\Phi}(u) := \operatorname*{argmin}_{v \in H} \{ \frac{1}{2} \|v - u\|^2 + \eta \Phi(v) \}. \tag{2.2} $$

The existence and uniqueness of the proximum means that the Moreau envelope is well-defined and proper. Moreover, it is convex and differentiable and its gradient is given by

$$ \nabla \left({}^{\eta}\Phi \right)(u) = \frac{1}{\eta} \big(u - \operatorname{prox}_{\eta\Phi}(u) \big), \tag{2.3} $$

see, e.g., [6, Theorem 5.2].

Let us now generalize the proximation problem (2.2). The main element we want to keep is the property that the functional to be minimized can be written as the sum of two operators with useful properties. So, most of the problems in this paper are of the form

$$ (P) \qquad \min_{u \in H_1} \big\{ \underbrace{g(u) + \Phi(Du)}_{:= \mathcal{F}_P(u)} \big\}, $$

where both functions $g : H_1 \to \overline{\mathbb{R}}$ and $\Phi : H_2 \to \overline{\mathbb{R}}$ are proper, convex and l.s.c., H_1 and H_2 are Hilbert spaces and $D : H_1 \to H_2$ is a bounded linear operator. We denote by I

the identity operator or the identity matrix in the finite-dimensional case. For $D = I$ and $g(u) := \frac{1}{2\eta}\|u - f\|^2$ in (P) we recover problem (2.1).

Observe that the objective function of (P) is neither strictly convex nor coercive in general so that without further assumptions a solution might not even exist or it might not be unique.

As we will see throughout this paper it is often useful to consider the dual problem of (P) or a combination of both. In the following, we will give, based on [29], a very brief motivation for duality in optimization theory. Let us first introduce the *Legendre conjugate* h^* of a proper, convex and l.s.c. functional $h : H \to \overline{\mathbb{R}}$ on a Hilbert space:

$$h^*(\phi) := \sup_{u \in H}\{\langle \phi, u\rangle - h(u)\}.$$

Note that h^* itself is proper, convex and l.s.c. and it holds for the biconjugate function that $h = h^{**}$. Now consider for some Hilbert space H the *perturbation* $\varphi : H_1 \times H \to \overline{\mathbb{R}}$ of $\mathcal{F}_P$ which means that

$$\varphi(\cdot, 0) = \mathcal{F}_P.$$

Furthermore, let us introduce the corresponding *value function*

$$v(b) = \inf_{u \in H_1} \varphi(u, b).$$

Note that $v(0)$ gives us the optimal value of (P). A short calculation shows that we get for the conjugate and the biconjugate of v the equalities

$$\begin{aligned} v^*(b) &= \varphi^*(0, b), \\ v^{**}(b) &= \sup_{\tilde{b} \in H}\{\langle b, \tilde{b}\rangle - \varphi^*(0, \tilde{b})\}. \end{aligned}$$

Motivated by the relation $v^{**} = v$ for v proper, convex and l.s.c. and the fact that $v(0) = \inf_{u \in H_1} \mathcal{F}_P(u)$ we define the *dual problem* as

$$v^{**}(0) = \sup_{\tilde{b} \in H}\{-\varphi^*(0, \tilde{b})\}. \tag{2.4}$$

We can also define a dual problem via a Lagrangian approach. Let us define the *duality Lagrangian* $L : H_1 \times H \times H \to \mathbb{R}$ as

$$L(u, \tilde{b}, b) := \langle \tilde{b}, b\rangle - \varphi_b^*(u, \tilde{b}),$$

where φ_b^* denotes the conjugate function with respect to the second variable, i.e.,

$$\varphi_b^*(u, \tilde{b}) := \sup_{b' \in H}\{\langle \tilde{b}, b'\rangle - \varphi(u, b')\}.$$

There is a direct connection between the duality Lagrangian L and the dual problem (D). Considering

$$\begin{aligned} \inf_{u \in H_1} L(u, \tilde{b}, b) &= \langle \tilde{b}, b\rangle - \sup_{u \in H_1, b' \in H}\{\langle 0, u\rangle + \langle \tilde{b}, b'\rangle - \varphi(u, b')\} \\ &= \langle \tilde{b}, x\rangle - \varphi^*(0, \tilde{b}) \end{aligned}$$

and consequently,

$$v^{**}(b) = \sup_{\tilde{b}\in H} \inf_{u\in H_1} L(u, \tilde{b}, b),$$

we recover the dual problem as

$$v^{**}(0) = \sup_{\tilde{b}\in H} \inf_{u\in H_1} L(u, \tilde{b}, 0). \tag{2.5}$$

The primal problem is obtained from the duality Lagrangian as follows: First, note that by definition

$$\sup_{\tilde{b}\in H} L(u, \tilde{b}, b) = \varphi_b^{**}(u, b).$$

So, assuming that $\varphi(u, \cdot)$ is proper, convex and l.s.c. we have

$$\sup_{\tilde{b}\in H} L(u, \tilde{b}, b) = \varphi(u, b)$$

and thus, the primal problem is given by

$$v(0) = \inf_{u\in H_1} \sup_{\tilde{b}\in H} L(u, \tilde{b}, 0). \tag{2.6}$$

So, in the Lagrangian formulation the primal and the dual problem (2.6) and (2.5), respectively, only differ in the order in which we optimize with respect to the two variables. Furthermore, observe that a tuple of solutions $(\hat{u}, \hat{b})$ of the primal and the dual problem, respectively, can be characterized as a saddle point of the duality Lagrangian $L(\cdot, \cdot, 0)$.

A useful perturbation of $\mathcal{F}_P$ is given by

$$\varphi(u, b) = g(u) + \Phi(Du + b).$$

It is straightforward to see that the corresponding dual problem of (P) is

$$(D) \qquad -\min_{b\in H_2} \big\{ \underbrace{g^*(-D^*b) + \Phi^*(b)}_{:=\mathcal{F}_D(b)} \big\}.$$

For example, observing that $(\frac{1}{2}\|\cdot - f\|^2)^*(\phi) = \frac{1}{2}\|\phi + f\|^2 - \frac{1}{2}\|f\|^2$, the dual problem of the proximation problem

$$\min_{u\in H} \big\{ \tfrac{1}{2}\|u - f\|_2^2 + \eta\Phi(u) \big\},$$

cf. (2.2), has the simple form

$$-\min_{b\in H} \big\{ \tfrac{1}{2}\|b - f\|^2 - \tfrac{1}{2}\|f\|^2 + \eta\Phi^*\big(\tfrac{1}{\eta}b\big) \big\} \tag{2.7}$$

and the unique solutions $\hat{u}$ and $\hat{b}$ of the primal and dual problem, respectively, are connected by $\hat{u} = f - \hat{b}$. Unless stated otherwise, we always assume in this paper that the primal and the dual solution $\hat{u}$ and $\hat{b}$, respectively, exist and that the duality gap is zero, i.e., the

extremal values $v(0)$ and $v^{**}(0)$ of (P) and (D) are the same. To put it another way, we suppose that there is a pair $(\hat{u}, \hat{d})$ which satisfies the *Karush-Kuhn-Tucker conditions*

$$0 \in \partial g(\hat{u}) + D^* \hat{b}, \tag{2.8}$$
$$0 \in -D\hat{u} + \partial \Phi^*(\hat{b}). \tag{2.9}$$

Recall that for a proper, convex function $g : H \to \overline{\mathbb{R}}$ the set-valued function $\partial g : H \to 2^H$ which is called the *subdifferential* of g is given by

$$\partial g(\tilde{u}) = \{v \in H \,:\, \langle v, u - \tilde{u}\rangle \leq g(u) - g(\tilde{u}),\ \forall u \in H\}. \tag{2.10}$$

An element of $\partial g(\tilde{u})$ is called a *subgradient* of g at the point $\tilde{u}$. Let us define the *core* of a subset C of a vector space as the set of points $u \in C$ with the property that for any direction v, $u + tv$ lies in C for all $0 \leq t < t_v$ where the positive value t_v is allowed to depend on v. Observe that the interior of a set is always contained in its core. We further assume in this paper that the following two inclusion which are often referred to as regularity conditions hold true:

$$0 \in \operatorname{core}(D \operatorname{dom} g - \operatorname{dom} \Phi), \tag{2.11}$$
$$0 \in \operatorname{core}(\operatorname{dom} g^* + D^* \operatorname{dom} \Phi^*). \tag{2.12}$$

The regularity conditions (2.11)-(2.12) imply via the Fenchel duality theorem, cf., e.g., [32, 183], that solutions of (P) and (D) exist and that the duality gap is zero. The converse, however, is not true in general. Observe that for every closed and convex set in a Banach space the core is equal to the interior of this set, cf., [31, 139]. Since we assume that g and Φ are proper, convex and l.s.c. the domains in (2.11)-(2.12) are closed and convex sets it follows that we could write (2.11)-(2.12) equivalently in terms of the interior of the two sets. The *relative interior* of a subset C of a vector space, denoted by $\operatorname{ri} C$, is defined as its interior within the affine hull, i.e., the smallest affine set containing C. In finite-dimensional spaces, (2.11)-(2.12) can be replaced by the following weaker assumptions, compare [181, Theorem 31.1]

$$\operatorname{ri}(D \operatorname{dom} g) \cap \operatorname{ri}(\operatorname{dom} \Phi) \neq \emptyset,$$
$$\operatorname{ri}(\operatorname{dom} g^*) \cap \operatorname{ri}(D^* \operatorname{dom} \Phi^*) \neq \emptyset.$$

Note that even in finite-dimensional spaces we cannot conclude from (2.11)-(2.12) that solutions of (P) and (D) exist and that the duality gap is zero.

Assuming that the conditions (2.11)-(2.12) are satisfied we have

$$\partial \mathcal{F}_P = \partial g + \partial(\Phi \circ D),$$
$$\partial \mathcal{F}_D = \partial(g^* \circ (-D^*)) + \partial \Phi^*.$$

Using the definition (2.10) of the subdifferential, we immediately see that $\hat{u}$ is a solution of (P) if and only if $0 \in \partial \mathcal{F}_P(\hat{u})$ and analogously for the dual problem. This inclusion is often referred to as *Fermat's rule*. So, we get

$$\boxed{\hat{u} = \operatorname{argmin}_{u \in H_1} \left\{ g(u) + \Phi(Du) \right\}} \Leftrightarrow \boxed{0 \in \partial g(\hat{u}) + \partial(\Phi \circ D)(\hat{u})} \tag{2.13}$$

$$\boxed{\hat{b} = \operatorname{argmin}_{b \in H_2} \left\{ g^*(-D^* b) + \Phi^*(b) \right\}} \Leftrightarrow \boxed{0 \in \partial(g^* \circ (-D^*))(\hat{b}) + \partial \Phi^*(\hat{b})} \tag{2.14}$$

Thus, both the primal and the dual problem can be written in the form of a common zero inclusion problem

$$0 \in A(\hat{p}) + B(\hat{p}), \tag{2.15}$$

where both A and B are set-valued operators.

The notions of convexity and lower semi-continuity are connected to set-valued operators via the concepts of monotonicity and maximal monotonicity. A set-valued operator $A : H \to 2^H$ is called *monotone* if

$$\langle u_1 - u_2, v_1 - v_2 \rangle \geq 0, \quad \forall\, v_1 \in A(u_1),\, v_2 \in A(u_2).$$

It is *maximal monotone* if in addition to being monotone the following property is satisfied: Let $u_1, v_1 \in H$ be fixed. It holds that $v_1 \in A(u_1)$ if for all pairs $(u_2, v_2) \in H \times H$ with $v_2 \in A(u_2)$ we have

$$\langle u_1 - u_2, v_1 - v_2 \rangle \geq 0.$$

Lemma 2.2.1 *If a function $g : H \to \overline{\mathbb{R}}$ is proper and convex, then the subdifferential ∂g is monotone. If g is also l.s.c., then ∂g is maximal monotone.*

Proof: [35, Chapter II]

Note that not all maximal monotone operators can be written as subdifferentials, see [87, p. 133] for a counterexample.

Theorem 2.2.2 *Let $A : H \to 2^H$ be a maximal monotone operator. Furthermore, let $(u^{(k)})_{k \in \mathbb{N}}$ and $(v^{(k)})_{k \in \mathbb{N}}$ be two sequences in H which converge weakly to $\hat{u}$ and $\hat{v}$, respectively. If $\operatorname{limsup}_{k \to \infty} \langle u_k, v_k \rangle \leq \langle \hat{u}, \hat{v} \rangle$ and $v^{(k)} \in A(u^{(k)})$ for all $k \in \mathbb{N}$, then it holds that $\hat{v} \in A(\hat{u})$.*

Proof: (See also [35, p. 27] in French.) Let $u, v \in H$ be any pair of elements such that $v \in A(u)$. Since A is monotone, we have for all $k \in \mathbb{N}$ that $\langle u - u^{(k)}, v - v^{(k)} \rangle \geq 0$. So, we obtain

$$\begin{aligned} 0 &\leq \operatorname*{limsup}_{k \to \infty} \langle u - u^{(k)}, v - v^{(k)} \rangle \\ &= \langle u, v \rangle - \langle \hat{u}, v \rangle - \langle u, \hat{v} \rangle + \operatorname*{limsup}_{k \to \infty} \langle u^{(k)}, v^{(k)} \rangle \\ &\leq \langle u, v \rangle - \langle \hat{u}, v \rangle - \langle u, \hat{v} \rangle + \langle \hat{u}, \hat{v} \rangle = \langle u - \hat{u}, v - \hat{v} \rangle. \end{aligned}$$

Since A is maximal monotone it follows that $\hat{v} \in A(\hat{u})$. □

It is clear that if in the above Theorem 2.2.2 either $(u^{(k)})_{k \in \mathbb{N}}$ or $(v^{(k)})_{k \in \mathbb{N}}$ converges strongly, then it holds that

$$\lim_{k \to \infty} \langle u_k, v_k \rangle = \langle \hat{u}, \hat{v} \rangle$$

and $\hat{v} \in A\hat{u}$ is true.

Let us now return to the proximation problem (2.2) and its dual (2.7). In terms of (2.13) and (2.14) this can be written as

$$f \in (I + \eta\partial\Phi)(\hat{u}), \quad f \in (I + \eta\partial\Phi^*(\tfrac{1}{\eta}\cdot))(\hat{b}). \tag{2.16}$$

Recall that $\hat{u}$ and $\hat{b}$ exist and are uniquely determined because Φ is proper and convex. This implies that the so-called *resolvent* $J_{\eta\partial\Phi} := (I + \eta\partial\Phi)^{-1}$ of Φ is a single-valued operator on H. The same holds true for the resolvent $J_{\eta\partial\Phi^*(\frac{1}{\eta}\cdot)}$. So, we can write (2.16) in the form

$$\hat{u} = J_{\eta\partial\Phi}(f), \quad \hat{b} = J_{\eta\partial\Phi^*(\frac{1}{\eta}\cdot)}(f).$$

The resolvents of more general operators will play a crucial role in the algorithms we present in Sections 2.3.2 and 2.3.3.

In the following two sections, we will describe methods to solve problems of the form (2.15), in particular the problems (P) and (D). We will start by studying operator splitting methods and then consider Bregman techniques.

2.3 Operator splittings

Operator splitting methods make use of the additive structure of the problem in the sense that they decompose the problem into subproblems corresponding to only one of the summands. Moreover, in these subproblems the specific structure of the individual terms can be taken into account.

2.3.1 Weak convergence of Picard iterations of averaged operators

In the this subsection, we present important definitions and general results on the convergence of fixed point iterations which will form the basis of our algorithms. In the convergence proofs of the various operator splittings methods presented in the next subsections we will apply convergence results on Picard iterations of so-called averaged operators. In connection with the common zero problem (2.15), for example, averaged operators of the form $T = J_{\eta A}(I - \eta B)$ or $T = J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I$ will appear.

Let $T : H \to H$ be an operator with fixed point set $\mathrm{Fix}(T)$. Then, we can try to find iteratively a fixed point by means of Picard iterations:

Algorithm (Picard Iterations)
Initialization: $u^{(0)} \in H$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$u^{(k+1)} = T(u^{(k)}). \tag{2.17}$$

Since we will eventually discretize the given problem we will focus on weak convergence of (2.17) in most parts of this chapter. We use the notation $\overset{w}{\to}$ for the weak limit.
A first idea is that convergence of (2.17) cannot arise if the operator increases the distance between two points. This idea motivates the following classical concept in optimization and fixed point theory. An operator T is *nonexpansive* if

$$\|T(u) - T(v)\| \leq \|u - v\|, \quad \forall u, v \in H.$$

However, this property alone does not yield convergence of an iterative method, e.g., translations and rotations in $\mathbb{R}^2$ are nonexpansive but the corresponding Picard iterations do not converge in general. In fact, in these examples T is *isometric*, i.e., one has

$$\|T(u) - T(v)\| = \|u - v\|, \quad \forall u, v \in H.$$

Hence, the only way for a sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by (2.17) with such an isometric operator T to converge is if $T(u^{(0)}) = u^{(0)}$ or in other words if we start at a fixed point. The following more restrictive concept avoids these problems in an obvious way. An operator T is *contractive* (also called a *contraction*) if it is *Lipschitz continuous* with Lipschitz constant $L \in (0, 1)$, i.e., if there exists an $L \in (0, 1)$ such that

$$\|T(u) - T(v)\| \leq L\|u - v\|, \quad \forall u, v \in H.$$

The *Banach fixed point theorem* guarantees that a contraction has a unique fixed point and that the Picard sequence (2.17) converges strongly to this fixed point for every initial element, see [131, Chapter 7] for more details and historical references.

However, in many applications the operators have useful properties but are not contractions. The main reason for this is that contractility is too restrictive in the sense that we often do not have a unique fixed point. Indeed, it is quite natural in many cases that the fixed point depends on the starting value $u^{(0)}$. Another reason to broaden the class of operators we want to study is that we often do not need strong convergence. So, we introduce the concept of averaged operators. A mapping $T : H \to H$ is called *averaged* if there exists a nonexpansive mapping R and a constant $\alpha \in (0, 1)$ such that

$$T = \alpha I + (1 - \alpha)R.$$

Historically, the concept of averaged mappings can be traced back to [138, 153, 189], where the name "averaged" was not used yet. The basic results on averaged operators collected in this sections can also be found, e.g., in [47, 76].

Example 2.3.1 *In $\mathbb{R}^2$ we consider the reflection operator*

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

Obviously, R is nonexpansive and not averaged and we only have convergence of $u^{(k+1)} = Ru^{(k)}$ if $u^{(0)} \in \mathrm{Fix}(R) = \mathrm{span}\{(1, 0)\}$. Now, for any $\alpha \in (0, 1)$ the averaged operator $T := \alpha I + (1 - \alpha)R$ has the same set of fixed points and for every $u^{(0)} = (u_1^{(0)}, u_2^{(0)}) \in \mathbb{R}^2$ the sequence $(T^k u^{(0)})_{k\in\mathbb{N}}$ converges to $(u_1^{(0)}, 0)$.

Clearly, an averaged operator is nonexpansive and in addition we have the following result.

Lemma 2.3.2 *A contractive operator $T : H \to H$ with Lipschitz constant $L < 1$ is averaged with respect to all parameters $\alpha \in (0, (1-L)/2]$.*

Proof: We define the operator $R := \frac{1}{1-\alpha}(T - \alpha I)$. It holds for all $u, v \in H$ that

$$\begin{aligned} \|Ru - Rv\| &= \frac{1}{1-\alpha}\|(T-\alpha I)u - (T-\alpha I)v\|, \\ &\leq \frac{1}{1-\alpha}\|Tu - Tv\| + \frac{\alpha}{1-\alpha}\|u - v\|, \\ &\leq \frac{L}{1-\alpha}\|u - v\| + \frac{\alpha}{1-\alpha}\|u - v\|. \end{aligned}$$

So, the operator R is nonexpansive if

$$\frac{L}{1-\alpha} + \frac{\alpha}{1-\alpha} \leq 1,$$

which is equivalent to $\alpha \leq (1-L)/2$. □

Note that from Example 2.3.1 it can be seen that the converse is not true.

Lemma 2.3.3 *Suppose that T is averaged with respect to the parameter $\alpha \in (0, 1)$. Then, it is also averaged with respect to any other parameter $\tilde{\alpha} \in (0, \alpha]$.*

Proof: By assumption, $T = \alpha I + (1-\alpha)R$ with R nonexpansive. We have

$$T = \tilde{\alpha} I + \underbrace{\big((\alpha - \tilde{\alpha})I + (1-\alpha)R\big)}_{:=\tilde{R}}$$

and for all $u, v \in H$ it holds that

$$\|\tilde{R}(u) - \tilde{R}(v)\| \leq (\alpha - \tilde{\alpha})\|u - v\| + (1-\alpha)\|R(u) - R(v)\| \leq (1 - \tilde{\alpha})\|u - v\|.$$

So, $\tilde{R}$ is nonexpansive (even a contraction). This concludes the proof. □

As we will see later in the context of the forward-backward splitting algorithm, the special case where the operator is averaged with respect to the parameter $\alpha = \frac{1}{2}$ is important. Such an operator is called *firmly nonexpansive*. Two equivalent definition of this property are given by

$$\|T(u) - T(v)\|^2 \leq \langle u - v, T(u) - T(v)\rangle, \quad \forall u, v \in H. \tag{2.18}$$

and

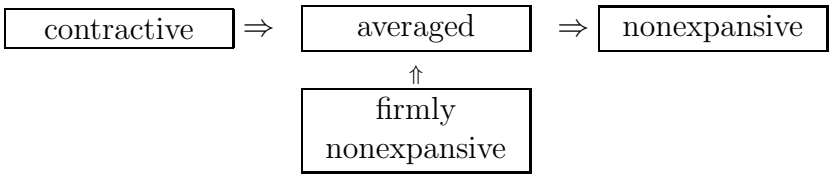
$$\|T(u) - T(v)\|^2 \leq \|u - v\|^2 - \|(I-T)(u) - (I-T)(v)\|^2, \quad \forall u, v \in H, \tag{2.19}$$

cf., e.g., [185]. We immediately see that T being firmly nonexpansive implies that it is single-valued and nonexpansive. It will be very useful in the following sections that the resolvents we are dealing with in this paper are indeed firmly nonexpansive.

Lemma 2.3.4 *Let $A : H \to 2^H$ be monotone. Then, the resolvent J_A is single-valued and nonexpansive. Moreover, A is maximal monotone if and only if J_A is firmly nonexpansive.*

Proof: See [158] and [7, pp. 107].

To summarize, it holds that



We will show now that the property of being averaged guarantees (weak) convergence of the sequence $(T^k(u^{(0)}))_{k\in\mathbb{N}}$. First, we prove the following lemma which we will often use in this paper.

Lemma 2.3.5 *Let $T_1 : H \to H$ and $T_2 : H \to H$ be two averaged operators. Then, the product $T_2 \circ T_1$ is also averaged.*

Proof: By assumption there exist nonexpansive operators S_1, S_2 and $\alpha_1, \alpha_2 \in (0,1)$ such that for all $u \in H$

$$\begin{aligned}
&T_2(T_1(u)) = (\alpha_2 I + (1-\alpha_2)S_2)(\alpha_1 I + (1-\alpha_1)S_1)(u)\\
&= \alpha_2\alpha_1 u + \alpha_2(1-\alpha_1)S_1(u) + (1-\alpha_2)\alpha_1 S_2(u) + (1-\alpha_2)(1-\alpha_1)S_2(S_1(u))\\
&= \alpha_2\alpha_1 u\\
&+(1-\alpha_1\alpha_2)\underbrace{\Big(\tfrac{1}{1-\alpha_1\alpha_2}\big(\alpha_1(1-\alpha_2)T_2(u) + \alpha_2(1-\alpha_1)T_1(u) + (1-\alpha_1)(1-\alpha_2)T_2(T_1(u))\big)\Big)}_{:=R}.
\end{aligned}$$

The product of two nonexpansive operators is clearly nonexpansive, so $T_2 \circ T_1$ is nonexpansive. Furthermore, the convex combination of nonexpansive operators is nonexpansive. The second summand on the right-hand side is indeed a convex combination of T_1, T_2 and $T_2 \circ T_1$ because $\alpha_1(1-\alpha_2)+\alpha_2(1-\alpha_1)+(1-\alpha_1)(1-\alpha_2) = 1-\alpha_1\alpha_2$. So, R is nonexpansive and thus $T_2 \circ T_1$ is averaged. □

For m averaged operators with parameters $\alpha_1, \ldots, \alpha_m$, the above Lemma 2.3.5 yields the parameter $\alpha = \Pi_{i=1}^m \alpha_i$ for the product mapping. This result can be sharpened in the sense of Lemma 2.3.3, see [76, Lemma 2.2], to the larger constant

$$\alpha = \frac{m}{m - 1 + \frac{1}{\max_{1\le i\le m}\alpha_i}}.$$

A mapping $T : H \to H$ is called *asymptotically regular* if it holds for all $u \in H$ that

$$T^{k+1}(u) - T^k(u) \to 0, \quad \text{for } k \to +\infty.$$

Observe that this property does not imply weak convergence, even boundedness cannot be guaranteed. In [37, Theorem 5], the following theorem is shown.

Theorem 2.3.6 *Let X be a uniformly convex Banach space and let $T : X \to X$ be an averaged mapping with respect to the nonexpansive mapping R and the parameter $\alpha \in (0,1)$. Assume that* $\mathrm{Fix}(T) \neq \emptyset$. *Then, T is asymptotically regular and* $\mathrm{Fix}(T) = \mathrm{Fix}(R)$.

Proof: The second part is clear: $T(u) = u \Leftrightarrow \alpha u + (1-\alpha)R(u) = u \Leftrightarrow (1-\alpha)R(u) = (1-\alpha)u$.
To prove the first part, we fix $\hat{u} \in \mathrm{Fix}(T)$ and write $u^{(k)} = T^k(u^{(0)})$ for some starting element $u^{(0)} \in X$. Since T is nonexpansive, i.e., $\|u^{(k+1)} - \hat{u}\| \leq \|u^{(k)} - \hat{u}\|$ we obtain

$$\lim_{k\to\infty} \|u^{(k)} - \hat{u}\| = d \geq 0.$$

We have to consider the case where $d > 0$. Let us introduce the variables $v^{(k)} = \frac{1}{d}(u^{(k)} - \hat{u})$ and $w^{(k)} = \frac{1}{d}(R(u^{(k)}) - \hat{u}) = \frac{1}{d}(R(u^{(k)}) - R(\hat{u}))$. Clearly, it holds that $\lim_{k\to\infty} \|v^{(k)}\| = 1$ and $\|w^{(k)}\| \leq \|v^{(k)}\|$. Since

$$\frac{1}{d}(u^{(k+1)} - \hat{u}) = \alpha v^{(k)} + (1-\alpha)w^{(k)},$$

we get

$$\frac{1}{d}\|u^{(k+1)} - \hat{u}\| \leq \alpha\|v^{(k)}\| + (1-\alpha)\|w^{(k)}\| \leq \|v^{(k)}\|.$$

The fact that $\|\frac{1}{d}(u^{(k+1)} - \hat{u})\| \to 1$ thus implies that also $\|w^{(k)}\| \to 1$ as $k \to \infty$. In summary, we have

$$\lim_{k\to\infty} \|v^{(k)}\| = \lim_{k\to\infty} \|w^{(k)}\| = 1$$

and

$$\lim_{k\to\infty} \|\alpha v^{(k)} + (1-\alpha)w^{(k)}\| = \lim_{k\to\infty} \|\frac{1}{d}(u^{(k+1)} - \hat{u})\| = 1.$$

So, by the uniform convexity of X we obtain

$$\lim_{k\to\infty} \|v^{(k)} - w^{(k)}\| = 0.$$

It follows that $\|u^{(k)} - R(u^{(k)})\| \to 0$ and since

$$\|u^{(k)} - u^{(k+1)}\| = \|u^{(k)} - \alpha u^{(k)} - (1-\alpha)R(u^{(k)})\| = (1-\alpha)\|u^{(k)} - R(u^{(k)})\|$$

it follows that $\|u^{(k)} - u^{(k+1)}\| \to 0$ as $k \to \infty$. □

Clearly, if H is a Hilbert space, then the above theorem can be applied because every Hilbert space is uniformly convex.

In order to show convergence of the iteration (2.17) we also need the following important result.

Theorem 2.3.7 *Let H be a Hilbert space , $V \subset H$ and let $T : V \to V$ be an asymptotically regular and nonexpansive operator with* $\mathrm{Fix}(T) \neq \emptyset$. *Then, for every $u^{(0)} \in V$, the sequence of Picard iterates $(u^{(k)})_{k\in\mathbb{N}}$ generated by* (2.17) *converges weakly to an element of* $\mathrm{Fix}(T)$.

This theorem was first proved by Opial [170, Theorem 1] based on results in [37]. In the appendix of [83] a shorter proof (following the lines of [170]) is given. Note that in [170] we find a remark concerning the generalization of this theorem to uniformly convex Banach spaces. It turns out that in this case one needs an additional condition on the structure of the space, cf., [170, p. 593]. In this paper, the proof follows from the proof of Theorem 2.3.11 as a special case, where $m = 1$ and $e^{(k)} = 0$ for all $k \in \mathbb{N}$.

Remark 2.3.8 *If we assume that the Hilbert space is finite-dimensional, the proof of Theorem 2.3.7 is very simple (we follow here* [37, Theorem 1, Theorem 2]*). First note that for any* $\hat{u} \in \mathrm{Fix}(T)$ *and any* $u^{(0)} \in H$ *we have* $\|T^{k+1}(u^{(0)}) - \hat{u}\| = \|T(T^k(u^{(0)})) - T(\hat{u})\| \leq \|T^k(u^{(0)}) - \hat{u}\|$*. So,* $(T^k(u^{(0)}))_{k\in\mathbb{N}}$ *is bounded. Now if* H *is finite-dimensional it follows that there exists a subsequence* $(u^{(k_l)})_{l\in\mathbb{N}}$ *which converges to some* $y \in H$*.*
If we can show that $y \in \mathrm{Fix}(T)$ *we are done because then* $\|T^k(u^{(0)}) - y\| \leq \|T^{k_l}(u^{(0)}) - y\|$ *for all* $k \geq k_l$ *and thus the whole sequence converges to* y*. Observe that since* T *is asymptotically regular it follows that* $T^{k_l+1}(u^{(0)}) - T^{k_l}(u^{(0)}) \to 0$*. We have* $T^{k_l+1}(u^{(0)}) - T^{k_l}(u^{(0)}) = (T - I)(T^{k_l}(u^{(0)}))$ *and since* $(T^{k_l}(u^{(0)}))_{l\in\mathbb{N}}$ *converges to* y *and* T *is continuous we get that* $(T - I)(y) = 0$*, i.e.,* $y \in \mathrm{Fix}(T)$*.*

If we combine the above Theorems 2.3.6 and 2.3.7 we obtain the following result.

Theorem 2.3.9 *Let* H *be a Hilbert space,* $V \subset H$ *and let* $T : V \to V$ *be an averaged mapping such that* $\mathrm{Fix}(T) \neq \emptyset$*. Then, for every* $u^{(0)} \in V$*, the sequence* $(T^k(u^{(0)}))_{k\in\mathbb{N}}$ *converges weakly to a fixed point of* T*.*

Remark 2.3.10 *Observe that the existence of fixed points is not guaranteed for an averaged mapping even in finite dimensions as the following example shows. Consider the translation* $Tx := x + 1$ *in* $\mathbb{R}$*. This mapping does not have any fixed points. However, it is averaged because it can be written as* $Tx = \frac{1}{2}x + \frac{1}{2}x + 1$ *and* $x \mapsto x + 2$ *is nonexpansive. For an overview of theorems concerning the existence of fixed points, see, e.g.,* [131, Chapters 4-6].

Since we want to split our original problem into subproblems that are easier to handle the averaged mapping T will in our examples are the sum of finitely many averaged mappings $T_1, \ldots, T_m$. In practice, we often cannot compute the application of the functionals $T_j : H \to H$ exactly. So, let $\left(e_j^{(k)}\right)_{j=1}^m \in H^m$ denote the tuple of errors which arise in each iteration $k \in \mathbb{N}$. This gives rise to the following inexact version of Picard iterations, see also [76]:

Algorithm (Inexact Picard Iterations)
Initialization: $u^{(0)} \in H$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$u^{(k+1)} := T_m(\ldots(T_1(u^{(k)}) + e_1^{(k)})\ldots) + e_m^{(k)}. \tag{2.20}$$

Theorem 2.3.11 *Let $V \subset H$ and let $T_j : V \to V$, $j = 1, \dots, m$, be nonexpansive operators. Assume that the right-hand side of* (2.20) *is well-defined and that the errors satisfy $\sum_{k=1}^{\infty} \|e_j^{(k)}\| < \infty$ for all $j = 1, \dots, m$. If for an initial element $u^{(0)} \in V$ it holds that $\lim_{k\to\infty} \|u^{(k+1)} - u^{(k)}\| = 0$ and if $\mathrm{Fix}(T_1 \cdots T_m) \neq \emptyset$ then* (2.20) *converges weakly to an element of $\mathrm{Fix}(T_1 \cdots T_m)$.*

It is important here that not only the normed errors have to go to zero but also that they have to be summable.

We will skip the short proof of Theorem 2.3.11 for the finite-dimensional case which is similar to the one given in Remark 2.3.8 for the exact version of this algorithm. Instead, we will now provide a proof for the infinite-dimensional setting based on Opial's proof, cf., [83, 170]. An inexact algorithm can also be found in [76] for an even more general setting, where the operators T_j themselves change to a certain degree in each iteration.

We want to borrow from [76] the notion of a *quasi-Féjer monotone sequence*. This is a sequence $(u^{(k)})_{k\in\mathbb{N}}$ in a Hilbert space with the property that

$$\|u^{(k+1)} - \hat{u}\| \leq \|u^{(k)} - \hat{u}\| + \epsilon_{k+1}, \quad \forall\, k \in \mathbb{N},\ \forall\, \hat{u} \in C, \tag{2.21}$$

where C is called the target set and $\epsilon_k \geq 0$ satisfies $\sum_{k=1}^{\infty} \epsilon_k < \infty$, compare [75]. In order to proof the main Lemma 2.3.14 about quasi-Féjer monotone sequences, we need the following two results. The first one generalizes Lemma B.5 in [83].

Lemma 2.3.12 *Let $(u^{(k)})_{k\in\mathbb{N}}$ be a quasi-Féjer monotone sequence with target set C. For each $\hat{u} \in C$ it holds that $(\|u^{(k)} - \hat{u}\|)_{k\in\mathbb{N}}$ converges.*

Proof: By definition, we have $\|u^{(k+1)} - \hat{u}\| \leq \|u^{(k)} - \hat{u}\| + \epsilon_{k+1}$. Let us write $d^{(k)} := \|u^{(k)} - \hat{u}\|$. Since $\sum_{k=1}^{\infty} \epsilon_k < \infty$ it follows that $(d^{(k)})_{k\in\mathbb{N}}$ is bounded. Now let $(d^{(n_k)})_{k\in\mathbb{N}}$ and $(d^{(l_k)})_{k\in\mathbb{N}}$ be two subsequences which converge to d_1 and d_2, respectively. Without loss of generality, assume that $d_1 > d_2$, so for every $\tau > 0$ small enough there exists a k_τ such that

$$d^{(n_{k_1})} - d^{(l_{k_2})} > \tau, \quad \forall k_1, k_2 \geq k_\tau. \tag{2.22}$$

Fix k_2 in (2.22) so that $\sum_{j=l_{k_2}}^{\infty} \epsilon_j < \tau$ and $k_2 \geq k_\tau$. Then, for all $k_1 > k_\tau$ with $n_{k_1} \geq l_{k_2}$ we have

$$d^{(n_{k_1})} \leq d^{(l_{k_2})} + \sum_{j=l_{k_2}}^{n_{k_1}-l_{k_2}} \epsilon_k \leq d^{(l_{k_2})} + \tau.$$

This yields a contradiction to (2.22) and we obtain $d_1 = d_2$. Thus, it follows that $(d^{(k)})_{k\in\mathbb{N}}$ converges. □

The second auxiliary lemma is Lemma B.2 of [83]:

Lemma 2.3.13 *Suppose that a sequence $(v^{(k)})_{k\in\mathbb{N}}$ in H converges weakly to $\hat{v}$. Then, for each $\tilde{v}$ with $\hat{v} \neq \tilde{v}$ we have*

$$\liminf_{k\to\infty} \|v^{(k)} - \tilde{v}\| > \liminf_{k\to\infty} \|v^{(k)} - \hat{v}\|.$$

Proof: We conclude that

$$\liminf_{k\to\infty} \|v^{(k)} - \tilde{v}\|^2 = \liminf_{k\to\infty} \big(\|v^{(k)} - \hat{v}\|^2 + 2\langle \underbrace{v^{(k)} - \hat{v}}_{\overset{w}{\to}0}, \hat{v} - \tilde{v}\rangle + \|\tilde{v} - \hat{v}\|^2\big).$$

□

We are now in the position to prove the following lemma about quasi-Féjer monotone sequences.

Lemma 2.3.14 *Let $C \subset H$ be a nonempty set in the Hilbert space H. If the sequence $(u^{(k)})_{k\in\mathbb{N}}$ is quasi-Féjer monotone with respect to C then it holds that $\mathcal{W}((u^{(k)})_{k\in\mathbb{N}})$, the set of all weak cluster points of $(u^{(k)})_{k\in\mathbb{N}}$, is nonempty and if $\mathcal{W}((u^{(k)})_{k\in\mathbb{N}}) \subset C$ then $(u^{(k)})_{k\in\mathbb{N}}$ converges weakly to a point in C.*

Proof: We can adapt [83, Theorem B.1] to our setting. First, we have to show that $\mathcal{W}((u^{(k)})_{k\in\mathbb{N}}) \neq \emptyset$. By Lemma 2.3.12, we have for any $\hat{u} \in C$ that

$$\limsup_{k\to\infty} \|u^{(k)}\| \leq \lim_{k\to\infty} \|u^{(k)} - \hat{u}\| + \|\hat{u}\| < \infty.$$

So, the sequence $(u^{(k)})_{k\in\mathbb{N}}$ is bounded and by the Banach-Alaoglu theorem it has a weak cluster point.
Now suppose u_1, $u_2 \in \mathcal{W}((u^{(k)})_{k\in\mathbb{N}}) \subset C$, i.e.,

$$u^{(n_k)} \overset{w}{\to} u_1, \quad u^{(l_k)} \overset{w}{\to} u_2.$$

Suppose that $u_1 \neq u_2$. It holds by Lemma 2.3.13 and Lemma 2.3.12 that

$$\lim_{k\to\infty} \|u^{(n_k)} - u_2\| > \lim_{k\to\infty} \|u^{(n_k)} - u_1\|,$$
$$\lim_{k\to\infty} \|u^{(l_k)} - u_1\| > \lim_{k\to\infty} \|u^{(l_k)} - u_2\|.$$

This yields a contradiction because

$$\lim_{k\to\infty} \|u^{(n_k)} - u_2\| = \lim_{k\to\infty} \|u^{(l_k)} - u_2\| = \lim_{k\to\infty} \|u^{(k)} - u_2\|$$

and similarly

$$\lim_{k\to\infty} \|u^{(l_k)} - u_1\| = \lim_{k\to\infty} \|u^{(n_k)} - u_1\| = \lim_{k\to\infty} \|u^{(k)} - u_1\|.$$

Consequently, we have $u_1 = u_2$ and thus $(u^{(k)})_{k\in\mathbb{N}}$ converges weakly the unique cluster point of this sequence. □

Applying Lemma 2.3.14 and Lemma 2.3.13 we can finally prove Theorem 2.3.11.

Proof of Theorem 2.3.11: Since $(u^{(k)})_{k\in\mathbb{N}}$ is by assumption quasi-Féjer monotone with respect to the nonempty target set $\mathrm{Fix}(T_m\cdots T_1)$ it remains to show that $\mathcal{W}((u^{(k)})_{k\in\mathbb{N}}) \subset \mathrm{Fix}(T_m\cdots T_1)$ in order to apply Lemma 2.3.14. So, let $u^{(n_k)} \overset{w}{\to} \hat{u}$ and consider

$$\liminf_{k\to\infty} \|u^{(n_k)} - T_m\cdots T_1(\hat{u})\| \leq \liminf_{k\to\infty} \|u^{(n_k)} - T_m\cdots T_1(u^{(n_k)})\| + \liminf_{k\to\infty} \|u^{(n_k)} - \hat{u}\|, \tag{2.23}$$

where we have used the triangle inequality and for the last term the fact that the operators $T_1,\ldots,T_m$ are nonexpansive. Note that by the definition of the sequence $(u^{(n_k)})_{k\in\mathbb{N}}$ and by the nonexpansivity of $T_1,\ldots,T_m$ it holds that

$$\begin{aligned} & \|u^{(n_k)} - T_m\cdots T_1(u^{(n_k)})\| \\ = \; & \|T_m(\ldots(T_1(u^{(n_k-1)}) + e_1^{(n_k-1)})\ldots) + e_m^{(n_k-1)} - T_m\cdots T_1(u^{(n_k)})\| \\ \leq \; & \|u^{(n_k-1)} - u^{(n_k)}\| + \sum_{i=1}^{m} \|e_i^{(n_k-1)}\|. \end{aligned}$$

Since we assume that $\|u^{(k+1)} - u^{(k)}\| \to 0$ as $k\to\infty$ and that $\lim_{k\to\infty}\sum_{i=1}^m \|e_i^{(k)}\| = 0$ we obtain

$$\lim_{k\to\infty} \|u^{(n_k)} - T_m\cdots T_1(u^{(n_k)})\| = 0. \tag{2.24}$$

So, the inequality (2.23) yields

$$\liminf_{k\to\infty} \|u^{(n_k)} - T_m\cdots T_1(\hat{u})\| \leq \liminf_{k\to\infty} \|u^{(n_k)} - \hat{u}\|$$

which means that by Lemma 2.3.13 we have $T_m\cdots T_1(\hat{u}) = \hat{u}$. Thus we have shown that $\mathcal{W}((u^{(k)})_{k\in\mathbb{N}}) \subset \mathrm{Fix}(T_m\cdots T_1)$ and the weak convergence follows from Lemma 2.3.14. □

So far we have assumed that $\lim_{k\to\infty}\|u^{(k+1)} - u^{(k)}\| = 0$. The next result generalizes Theorem 2.3.6 and shows that this condition is fulfilled for averaged operators.

Theorem 2.3.15 *Let X be a uniformly convex Banach space and consider a subset $V \subset X$ as well as the averaged mappings $T_i : V \to V$, $i = 1,\ldots,m$, with $\mathrm{Fix}(T_m\cdots T_1) \neq \emptyset$. Assume that the right-hand side of (2.20) is well-defined and that the errors satisfy $\sum_{k=1}^{\infty}\|e_j^{(k)}\| < \infty$ for $j = 1,\ldots,m$. Then, for each initial element $u^{(0)} \in V$ the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by (2.20) has the property that $\|u^{(k+1)} - u^{(k)}\| \to 0$ as $k\to\infty$.*

Proof: Consider an element $\hat{u} \in \mathrm{Fix}(T_m\cdots T_1)$. We know by Lemma 2.3.12 that

$$\lim_{k\to\infty} \|u^{(k)} - \hat{u}\| = d \geq 0.$$

Suppose that $d > 0$. Let R be a nonexpansive mapping such that $T_m\cdots T_1$ can be written as

$$T_m\cdots T_1 = \alpha I + (1-\alpha)R,$$

for some $\alpha \in (0,1)$. Define $v^{(k)} = \frac{1}{d}(u^{(k)} - \hat{u})$ and $w^{(k)} = \frac{1}{d}(Ru^{(k)} - \hat{u})$. The operator R is nonexpansive and by Theorem 2.3.6 the element $\hat{u}$ is also a fixed point of R. So, we have $\|v^{(k)}\| \to 1$ as $k \to \infty$ and $\|w^{(k)}\| \leq \|v^{(k)}\|$. It holds that

$$\tfrac{1}{d}(T_m \cdots T_1 u^{(k)} - \hat{u}) = \alpha v^{(k)} + (1-\alpha) w^{(k)}, \tag{2.25}$$

so that we get

$$\tfrac{1}{d}\|T_m \cdots T_1(u^{(k)}) - \hat{u}\| \leq \alpha \|v^{(k)}\| + (1-\alpha)\|w^{(k)}\| \leq \|v^{(k)}\|. \tag{2.26}$$

Using (2.24) and the triangle inequality to get

$$\tfrac{1}{d}\|T_m \cdots T_1(u^{(k)}) - \hat{u}\| \leq \underbrace{\tfrac{1}{d}\|T_m \cdots T_1(u^{(k)}) - u^{(k+1)}\|}_{\to 0} + \underbrace{\tfrac{1}{d}\|u^{(k+1)} - \hat{u}\|}_{\to 1}, \tag{2.27}$$

$$\underbrace{\tfrac{1}{d}\|u^{(k+1)} - \hat{u}\|}_{\to 1} \leq \underbrace{\tfrac{1}{d}\|T_m \cdots T_1(u^{(k)}) - u^{(k+1)}\|}_{\to 0} + \tfrac{1}{d}\|T_m \cdots T_1(u^{(k)}) - \hat{u}\|, \tag{2.28}$$

we obtain

$$\lim_{k\to\infty} \tfrac{1}{d}\|T_m \cdots T_1(u^{(k)}) - \hat{u}\| = 1.$$

By (2.25) and (2.26) it follows that

$$\lim_{k\to\infty} \|\alpha v^{(k)} + (1-\alpha) w^{(k)}\| = 1 \text{ and } \lim_{k\to\infty} \|w^{(k)}\| = 1.$$

So, by the uniform convexity of our space X we get

$$\lim_{k\to\infty} \|v^{(k)} - w^{(k)}\| = 0.$$

It follows that $\|u^{(k)} - R(u^{(k)})\| \to 0$ and from $\|u^{(k)} - \alpha u^{(k)} - (1-\alpha)R(u^{(k)})\| = (1-\alpha)\|u^{(k)} - R(u^{(k)})\|$ we immediately conclude that $\|u^{(k)} - T_m \cdots T_1(u^{(k)})\| \to 0$ as $k \to \infty$. Since

$$\|u^{(k+1)} - u^{(k)}\| \leq \|u^{(k+1)} - T_m \cdots T_1(u^{(k)})\| + \|T_m \cdots T_1(u^{(k)}) - u^{(k)}\|$$

it follows that $\|u^{(k+1)} - u^{(k)}\| \to 0$ as $k \to \infty$. □

Hence, analogously to Theorem 2.3.9 we obtain the following result.

Theorem 2.3.16 *Let H be a Hilbert space, $V \subset H$ and let $T_j : V \to V$, $j = 1, \ldots, m$, be averaged operators such that* $\mathrm{Fix}(T_m \cdots T_1) \neq \emptyset$. *Then, for every $u^{(0)} \in V$ the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by (2.20) converges weakly to a fixed point of the operators $T_m \cdots T_1$ if $\sum_{k=1}^{\infty} \|e_j^{(k)}\| < \infty$ for $j = 1, \ldots, m$.*

Generalization of (2.17) and (2.20) to the case where the averaged operators change in each iteration can be found in [126, 217, 227, 229, 233].

2.3.2 Forward–backward splitting

In this subsection we want to apply the convergence results of the preceding section to a first splitting algorithm for solving (2.15), the forward-backward splitting method which can be traced back to [144, 171]. The idea of the forward-backward splitting algorithm is that for any constant $\eta > 0$ we have

$$\begin{aligned} 0 \in A(\hat{u}) + B(\hat{u}) &\Leftrightarrow \hat{u} - \eta B(\hat{u}) \in \hat{u} + \eta A(\hat{u}) \\ &\Leftrightarrow \hat{u} \in J_{\eta A}(I - \eta B)(\hat{u}), \end{aligned} \tag{2.29}$$

where $\hat{u} - \eta B(\hat{u}) \in \hat{u} + \eta A(\hat{u})$ means that there exists an element $y \in B(\hat{u})$ such that $\hat{u} - \eta y \in \hat{u} + \eta A(\hat{u})$. This leads to the following result:

Theorem 2.3.17 (**Forward-backward splitting algorithm (FBS)**) *Let H be a Hilbert space. Suppose that $A : H \to 2^H$ is maximal monotone and $B : H \to H$ is a monotone operator such that βB is firmly nonexpansive for some $\beta > 0$. Furthermore, assume that a solution of* (2.15) *exists. Then, for every starting element $u^{(0)}$ and $\eta \in (0, 2\beta)$ the* forward-backward splitting algorithm

$$u^{(k+1)} = J_{\eta A}(I - \eta B)(u^{(k)}), \tag{2.30}$$

converges weakly to an element of the set of solutions $(A + B)^{-1}(\{0\})$.

Proof: Since A is maximal monotone it follows by Lemma 2.3.4 that for all $\eta > 0$ the resolvent $J_{\eta A}$ is single-valued and firmly nonexpansive. Furthermore, it holds that if βB is firmly nonexpansive, then $I - \eta B$ is averaged with parameter $\frac{\eta}{2\beta}$ for $\eta \in (0, 2\beta)$ which can be seen as follows: Assume that βB is firmly nonexpansive, then by definition there is a nonexpansive operator R s.t.

$$B = \tfrac{1}{2\beta}(I + R).$$

It follows that

$$I - \eta B = I - \tfrac{\eta}{2\beta}(I + R) = (1 - \tfrac{\eta}{2\beta})I + \tfrac{\eta}{2\beta}(-R).$$

So, both $J_{\eta A}$ and $I - \eta B$ are averaged if $\eta \in (0, 2\beta)$ and, consequently, the product $J_{\eta A}(I - \eta B)$ is averaged. Assuming that $\mathrm{Fix}(J_{\eta A}(I - \eta B)) \neq \emptyset$ we can apply Theorem 2.3.9 to conclude that for each starting point $u^{(0)}$ the iteration (2.30) converges weakly to a point $\hat{u} \in \mathrm{Fix}(J_{\eta A}(I - \eta B))$. Clearly, by (2.29), $\hat{u} \in \mathrm{Fix}(J_{\eta A}(I - \eta B))$ is equivalent to $0 \in A(\hat{u}) + B(\hat{u})$ and we are done. □

For a different proof, see, e.g., [211]. Various applications of the forward-backward splitting algorithm are reported in [80]. The name forward-backward splitting is due to the fact that in every iteration of (2.30) we perform a forward (explicit) step with respect to the operator B and a backward (implicit) step on A. Intuitively speaking, the conditions on the operator B allow us to do a simple forward step whereas A is a more general operator and we have to perform a backward step. The constant η can be interpreted as the step length.

Remark 2.3.18 *Clearly, based on Theorem 2.3.16 a similar result as Theorem 2.3.17 can be derived for the inexact case, i.e., when the application of the two averaged operators $J_{\eta A}$ and $I - \eta B$ introduces errors in each iteration.*

Recall that in Theorem 2.3.17 the operator B must be firmly nonexpansive. In many cases, the following result can be used to verify this property.

Theorem 2.3.19 *Let H be a Hilbert space and let the function $\mathcal{F}$ on H be real-valued, convex and Fréchet differentiable. If $\nabla\mathcal{F}$ is Lipschitz continuous with Lipschitz constant $\alpha > 0$, then $\frac{1}{\alpha}\nabla\mathcal{F}$ is firmly nonexpansive.*

This result goes back to [10, Corollaire 10] and a shorter proof was recently provided in [16]. Note that if $\frac{1}{\alpha}\nabla\mathcal{F}$ is firmly nonexpansive then $\nabla\mathcal{F}$ is also called $\frac{1}{\alpha}$-*cocoercive*.

2.3.3 Douglas–Rachford splitting

Note that in the foward-backward splitting algorithm the operator B has to be single-valued and we do not make use of the resolvent of B. This is different for the Douglas-Rachford splitting algorithm, cf., [92, 144], which is based on another fixed point equation. To introduce it, we first note that if B is single-valued we can rewrite the fixed point relation (2.29) as follows

$$\begin{aligned} 0 \in A(\hat{u}) + B(\hat{u}) \quad &\Leftrightarrow \quad \hat{u} \in J_{\eta A}(I - \eta B)(\hat{u}) \\ &\Leftrightarrow \quad \hat{u} + \eta B(\hat{u}) \in J_{\eta A}(I - \eta B)(\hat{u}) + \eta B(\hat{u}) \qquad (2.31) \\ &\Leftrightarrow \quad \hat{u} \in J_{\eta B}\big(J_{\eta A}(I - \eta B)(\hat{u}) + \eta B(\hat{u})\big). \qquad (2.32) \end{aligned}$$

If B is set-valued the Picard iterations

$$u^{(k+1)} \in J_{\eta B}\big(J_{\eta A}(I - \eta B)(u^{(k)}) + \eta B(u^{(k)})\big) \qquad (2.33)$$

corresponding to (2.32) are called the "loose" Douglas-Rachford splitting algorithm, cf. [94]. In general, the algorithm (2.33) does not converge to a solution of (2.15). In particular, we cannot guarantee that (2.31) follows from (2.32). However, one can choose the element of $\eta B(u^{(k)})$ in a special way to obtain a convergent algorithm. To this end, consider a fixed point $\hat{t} = Q(\hat{t})$ of the operator $Q : H \to H$ defined by

$$Q = J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I.$$

For such a fixed point $\hat{t}$ we define $\hat{u} := J_{\eta B}(\hat{t})$ and thus $\hat{\xi} := \hat{t} - \hat{u}$ lies in $\eta B(\hat{u})$. With this choice $\hat{\xi} \in \eta B(\hat{u})$ the element $\hat{u}$ is indeed a solution of (2.15). To see this, observe that with the above definition of $\hat{t}, \hat{u}$ and $\hat{\xi}$ the inner part of the right-hand side of (2.32) is equivalent to $Q(\hat{t})$ since

$$\begin{aligned} Q(\hat{t}) &= J_{\eta A}(2J_{\eta B}(\hat{t}) - \hat{t}) - J_{\eta B}(\hat{t}) + \hat{t} \\ &= J_{\eta A}(\hat{u} - \hat{\xi}) + \hat{\xi}. \qquad (2.34) \end{aligned}$$

By using $\hat{t} = Q(\hat{t})$ together with (2.34), we now recover the relation (2.31):

$$\hat{t} = Q(\hat{t}) \quad \Leftrightarrow \quad \hat{u} + \hat{\xi} = J_{\eta A}(\hat{u} - \hat{\xi}) + \hat{\xi}$$

and we can finally conclude that

$$\hat{u} \in J_{\eta A}(I - \eta B)(\hat{u}) \Leftrightarrow 0 \in A(\hat{u}) + B(\hat{u}).$$

We thus have to find a way to compute a fixed point of Q. As Lemma 2.3.20 below shows, the operator Q is firmly nonexpansive, i.e., it is averaged. Hence, we can conclude from Theorem 2.3.9 that for any starting element $t^{(0)} \in H$ and any step length $\eta > 0$ the Picard iterations

$$t^{(k+1)} = Q(t^{(k)}) = J_{\eta A}(2J_{\eta B}(t^{(k)}) - t^{(k)}) - J_{\eta B}(t^{(k)}) + t^{(k)} \tag{2.35}$$

converge weakly to an element $\hat{t}$ if a fixed points of Q exists.

Let us define the *reflection operator* $R_{\eta A} := 2J_{\eta A} - I$, where A is any maximal monotone operator and $\eta > 0$. We have:

Lemma 2.3.20 *Let A, B be two maximal monotone operators and $\eta > 0$ a fixed parameter. Then, it holds that the operator $Q := J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I$ is firmly nonexpansive with respect to the nonexpansive mapping $R_{\eta A}R_{\eta B}$.*

Proof: (Compare [144, Lemma 1] and [76, Lemma 2.6].) We know by Lemma 2.3.4 that the resolvent $J_{\eta A}$ is firmly nonexpansive which means by definition that $2J_{\eta A} - I$ is nonexpansive. The same holds true for $J_{\eta B}$. So, $R_{\eta A}R_{\eta B}$ is nonexpansive and we have

$$I + R_{\eta A}R_{\eta B} = 2J_{\eta A}(2J_{\eta B} - I) - (2J_{\eta B} - I) + I = 2(J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I).$$

This implies that

$$J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I = \tfrac{1}{2}(I + R_{\eta A}R_{\eta B}), \tag{2.36}$$

which concludes the proof. □

In summary, we have proved the following theorem, compare [76, 144].

Theorem 2.3.21 (**Douglas-Rachford splitting algorithm (DRS)**) *Let $A, B : H \to 2^H$ be maximal monotone operators and assume that a solution of* (2.15) *exists. Then, for any initial elements $t^{(0)}$ and $u^{(0)}$ and any $\eta > 0$, the following* (tight) Douglas-Rachford splitting algorithm *converges weakly to an element $\hat{t}$:*

$$\begin{aligned} t^{(k+1)} &= J_{\eta A}(2u^{(k)} - t^{(k)}) + t^{(k)} - u^{(k)}, \\ u^{(k+1)} &= J_{\eta B}(t^{(k+1)}). \end{aligned}$$

Furthermore, it holds that $\hat{u} := J_{\eta B}(\hat{t})$ satisfies $0 \in A(\hat{u}) + B(\hat{u})$. If H is finite-dimensional, then the sequence $\left(u^{(k)}\right)_{k\in\mathbb{N}}$ converges to $\hat{u}$.

Note that $J_{\eta B}$ being nonexpansive implies that it is continuous. Hence, if $(t^{(k)})_{k\in\mathbb{N}}$ is strongly convergent, then so is $(u^{(k)})_{k\in\mathbb{N}}$. In particular, we have convergence in finite-dimensional spaces.
Observe that for the above Douglas-Rachford algorithm we apply both the resolvents of A and of B. In contrast to that, the forward-backward splitting algorithm uses the resolvent of A, but the operator B is applied directly.
Generally, the convergence of the sequence $(u^{(k)})_{k\in\mathbb{N}}$ is not guaranteed in the infinite-dimensional setting. However, two properties of $(u^{(k)})_{k\in\mathbb{N}}$ can easily be established, compare [144, Proposition 2].

- $\lim_{k\to\infty} \|u^{(k+1)} - u^{(k)}\| = 0$
- The sequence $(u^{(k)})_{k\in\mathbb{N}}$ is bounded.

To see the first part, recall that we know by Lemma 2.3.20 that the operator $J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I$ is averaged. By (2.35) and Theorem 2.3.6 we thus have $t^{(k+1)} - t^{(k)} \to 0$ for $k \to \infty$. Furthermore, the operator $J_{\eta B}$ is nonexpansive which yields

$$\|u^{(k+1)} - u^{(k)}\| \leq \|J_{\eta B}(t^{(k+1)}) - J_{\eta B}(t^{(k)})\| \leq \|t^{(k+1)} - t^{(k)}\| \to 0.$$

For the second property, we can use that every weakly convergent sequence in a Banach space is bounded, so that $(t^{(k)})_{k\in\mathbb{N}}$ is bounded. Using again the nonexpansivity of $J_{\eta B}$, we obtain for some constant K that

$$\|u^{(k)} - \hat{u}\| \leq \|t^{(k)} - \hat{t}\| \leq \|t^{(k)}\| + \|\hat{t}\| \leq K, \quad \forall k \in \mathbb{N}.$$

For more general results on the convergence properties of $(u^{(k)})_{k\in\mathbb{N}}$, see [144, pp. 969–970].

Remark 2.3.22 *In the original paper* [92]*, the Douglas-Rachford splitting method was proposed for the numerical solution partial differential equation, i.e., for solving systems of linear equations. More on this application can also be found in* [213].

Remark 2.3.23 *For splitting algorithms to minimize a functional which is the sum of more than two functions, see Section 2.4.3 as well as, e.g.* [77, 79].

2.3.4 Other operator splittings

Backward-backward splitting We want to mention this method only briefly here since it does not yield fast algorithms for the minimization problems we consider in the following chapters. The *backward-backward splitting algorithm* [1, 17, 76, 144] has the form

$$u^{(k+1)} = J_{\eta A} J_{\eta B}(u^{(k)}), \tag{2.37}$$

where A and B are two maximal monotone operators and η is a positive constant. Clearly, the composition of two resolvents yields an averaged operator, so that by Theorem 2.3.9 for each starting element $u^{(0)}$, the algorithm (2.37) converges weakly to a fixed point of the

operator $J_{\eta A}J_{\eta B}$ if such a fixed point exists. However, the set $\mathrm{Fix}(J_{\eta A}J_{\eta B})$ does in general not coincide with the set $(A+B)^{-1}(\{0\})$ we are interested in. Instead, we have

$$\begin{aligned}\hat{u} = J_{\eta A}J_{\eta B}(\hat{u}) &\Leftrightarrow (I+\eta A)(\hat{u}) = J_{\eta B}(\hat{u}) \\ &\Leftrightarrow 0 \in A(\hat{u}) + \underbrace{\frac{1}{\eta}(I - J_{\eta B})}_{=:{}^{\eta}B}(\hat{u}).\end{aligned} \tag{2.38}$$

The mapping ${}^{\eta}B = \frac{1}{\eta}(I - J_{\eta B})$ is called the *Yosida approximant* with respect to B and the parameter η.

In the case where A and B are the subdifferentials of proper, convex and l.s.c. functionals ηg_1 and g_2, the problem (2.38) which is solved by the backward-backward splitting algorithm (2.37)

$$u^{(k+1)} = \mathrm{prox}_{\eta^2 g_1}\mathrm{prox}_{\eta g_2}u^{(k)}, \tag{2.39}$$

can be interpreted in a straightforward way. Recalling the definition of the gradient of the Moreau envelope given in (2.3), we see that the gradient of the Moreau envelope coincides with the Yosida approximant of the corresponding subdifferential, i.e., we have

$${}^{\eta}\partial g_2 = \nabla({}^{\eta}g_2). \tag{2.40}$$

It thus follows that the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by (2.39) converges weakly to a solution $\hat{u}$ of

$$\begin{aligned}& 0 \in \eta\partial g_1(\hat{u}) + \nabla({}^{\eta}g_2)(\hat{u}) \\ \Leftrightarrow\ & \hat{u} = \operatorname*{argmin}_{u\in H}\{\eta g_1(u) + \min_{v\in H}\{\frac{1}{2}\|v-u\|^2 + \eta g_2(v)\}\} \\ \Leftrightarrow\ & \hat{u} = \operatorname*{argmin}_{u\in H}\{\min_{v\in H}\{\frac{1}{2}\|v-u\|^2 + \eta g_1(u) + \eta g_2(v)\}\}\end{aligned} \tag{2.41}$$

So, the backward-backward method does not yield an exact solution of the problem $\mathrm{argmin}_u\{g_1(u)+g_2(u)\}$ but of a "regularized" version this problem which involves the penalty or coupling term $\frac{1}{2}\|u-v\|_2^2$. Clearly, we can use the parameter η to steer how close we get to the original solution. Observe that in (2.41) the minimum with respect to v is attained, too. This can be seen already from (2.38) noting the fact that $u^{(k)}$ converges $J_{\eta B}$ is continuous. We thus have in (2.41) that

$$\begin{aligned}& \hat{v} = \operatorname*{argmin}_{v\in H}\{\frac{1}{2}\|v-\hat{u}\|^2 + \eta g_2(v)\} \\ \Leftrightarrow\ & \hat{v} = \mathrm{prox}_{\eta g_2}\hat{u}.\end{aligned}$$

Peaceman-Rachford splitting The loose version of the *Peaceman-Rachford splitting algorithm* to find an element of $(A+B)^{-1}(\{0\})$ can be written as

$$u^{(k+1)} \in J_{\eta B}(I-\eta A)J_{\eta A}(I-\eta B)(u^{(k)}),$$

see [76, 94, 144, 172]. Again, A and B are maximal monotone operators. This can roughly be interpreted as the composition of two forward-backward splitting steps in each iteration, where the roles of the operators A and B change. A tight variant of this algorithm was proposed in [144] in terms of the reflection operator

$$t^{(k+1)} = R_{\eta A} R_{\eta B}(t^{(k)}). \tag{2.42}$$

If the iteration (2.42) converges to a fixed point $\hat{t}$ of $R_{\eta A} R_{\eta B}$, then $J_{\eta B}(\hat{t})$ is a solution of (2.15). This can be seen as follows. It holds that

$$\hat{t} = R_{\eta A} R_{\eta B}(\hat{t}) \;\Leftrightarrow\; \hat{t} = \tfrac{1}{2}(I + R_{\eta A} R_{\eta B})(\hat{t})$$

and by (2.36) we conclude that the fixed points of $R_{\eta A} R_{\eta B}$ coincide with those of the operator $J_{\eta A}(2J_{\eta B} - I) - J_{\eta B} + I$ used in the Douglas-Rachford algorithm. In contrast to the Douglas-Rachford splitting method, we cannot conclude the convergence of the Peaceman-Rachford splitting algorithm from the results of Section 2.3.3. We know from Lemma 2.3.20 that $R_{\eta A} R_{\eta B}$ is nonexpansive. In general, however, it is not averaged so that further assumptions on the operators A and B are necessary to guarantee convergence. Unfortunately, the conditions given in [76, 144] to guarantee convergence of the sequence $(t^{(k)})_{k\in\mathbb{N}}$ do not hold for the applications we consider in Chapters 3 and 4. This is confirmed experimentally for the Rudin-Osher-Fatemi model, i.e., cases did arise where in a reasonable timespan convergence could not be observed. Even when convergence occurred the algorithm was much slower than the corresponding forward-backward and Douglas-Rachford splitting algorithms.

2.4 Bregman methods

Recall that the idea behind the operator splitting methods considered in the preceding section is to decompose an original problem

$$\min_{u\in H} \mathcal{F}(u)$$

into easier parts, e.g., we compute the resolvent of just one summand of the problem at a time. The price we have to pay is that we obtain an iterative algorithm, i.e., we have to repeat these steps possibly many times. Bregman methods, too, are iterative algorithms which solve in each step a "nicer" problem.

2.4.1 Bregman proximal point method

We start by considering the general Bregman proximal point method and then deal with the special case of the classical proximal point algorithm for (P) and (D). The new problems we consider in Bregman iterative methods are constructed in the following way. We add to the original objective functional $\mathcal{F}$ a so-called "cost-to-move" term. This term penalizes

in some sense the distance between two iterates. The distance functions we use here are defined by means of *Bregman functions* $\varphi : H \to \overline{\mathbb{R}}$, cf. [34, 53]. We make the following assumptions on the space H and the functions φ:

A1: H is finite-dimensional,

A2: $\operatorname{dom}\varphi = H$,

A3: φ is l.s.c. and strictly convex.

The corresponding *Bregman distance* $D_\varphi^{(p)} : H \times H \to \overline{\mathbb{R}}$ is then given by

$$D_\varphi^{(p)}(u, v) = \varphi(u) - \varphi(v) - \langle p, u - v\rangle,$$

with $p \in \partial\varphi(v)$. Notice that if φ is smooth, then the Bregman distance can be interpreted as subtracting the first-order Taylor expansion of φ from the function φ itself. Based on these definitions we can follow [55] and define the *Bregman proximal point* (BPP) *algorithm* applied to a proper, convex and l.s.c. function $\mathcal{F} : H \to \overline{\mathbb{R}}$ as follows:

Algorithm (Bregman proximal point algorithm (BPP))
Initialization: $u^{(0)}, p^{(0)}$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} u^{(k+1)} &= \operatorname*{argmin}_{u\in H}\{\tfrac{1}{\gamma} D_\varphi^{(p^{(k)})}(u, u^{(k)}) + \mathcal{F}(u)\}, \\ p^{(k+1)} &\in \partial\varphi(u^{(k+1)}). \end{aligned} \tag{2.43}$$

We will often use the notation $D_\varphi^{(k)} := D_\varphi^{(p^{(k)})}$. Clearly, we have to make assumptions on φ and $D_\varphi^{(p)}$ for this algorithm to converge to a minimizer of $\mathcal{F}$. There are different approaches to do this. In contrast to the classical setting [55, 54, 69, 95, 103, 132, 136], we do not need that φ is differentiable. However, we restrict our attention to the case where the so-called zone of the Bregman function φ, i.e., its domain $\operatorname{dom}\varphi$, consists of the whole space H. For a more general setting, see [136]. We define the following functions

$$D_\varphi^{'}(u, v) = \varphi(u) - \varphi(v) - \sigma_{\partial\varphi(v)}(u - v), \tag{2.44}$$

$$D_\varphi^{\sharp}(u, v) = \varphi(u) - \varphi(v) + \sigma_{\partial\varphi(v)}(v - u), \tag{2.45}$$

where $\sigma_{\partial\varphi(v)}$ denotes the *support function* of the set $\partial\varphi(v)$, i.e.,

$$\sigma_{\partial\varphi(v)}(d) = \sup\{\langle w, d\rangle \,:\, w \in \partial\varphi(v)\}.$$

By [181, Theorem 23.4] it holds that

$$D_\varphi'(u, v) = \varphi(u) - \varphi(v) - D\varphi(v)(u - v),$$

where $D\varphi(v)(d)$ denotes the (right) directional derivative of φ at v with respect to the direction $d \in H$, i.e.,

$$D\varphi(v)(d) = \lim_{\alpha \to 0+} \frac{\varphi(v + \alpha d) - \varphi(v)}{\alpha}.$$

Observe that for any $u, v \in H$ and any subgradient $p \in \partial\varphi(v)$ we have

$$0 \leq D'_\varphi(u, v) \leq D^{(p)}_\varphi(u, v) \leq D^\sharp_\varphi(u, v). \tag{2.46}$$

All these functionals can be interpreted as generalized distance functions in the sense that they are nonnegative and attain, due to the strict convexity of φ, the value zero if and only if $u = v$. In general, however, they are not symmetric and the triangle inequality does not hold true. In Fig. 2.1 we show an example of a nonsmooth Bregman function φ and the corresponding distances D'_φ and $D^\sharp_\varphi$.

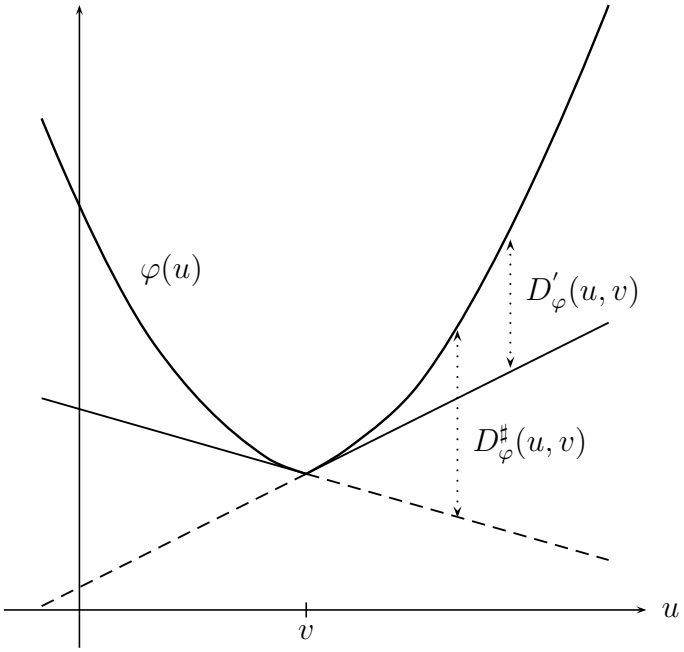


Figure 2.1: Example of a Bregman function φ on $\mathbb{R}$ and the corresponding distance functions $D^\sharp_\varphi(u, v)$ (distance between φ and solid lines) and $D'_\varphi(u, v)$ (distance between φ and dashed lines) for some fixed point v.

In order to prove the convergence of the Bregman proximal point algorithm we need the following lemmas.

Lemma 2.4.1 *Let H be an infinite-dimensional Hilbert space and let $f : H \to \overline{\mathbb{R}}$ be a proper, convex and l.s.c. function which is continuous on its domain. Assume $(u^{(k)})_{k\in\mathbb{N}}$ and $(d^{(k)})_{k\in\mathbb{N}}$ are strongly convergent sequences with limits $\hat{u}$ and $\hat{d}$, respectively, so that $u^{(k)}, u^{(k)} + d^{(k)} \in \mathrm{dom} f$, for all $k \in \mathbb{N}$ and $\hat{u}, \hat{u} + \hat{d} \in \mathrm{dom} f$. Then, it holds that*

$$\limsup_{k\to\infty} Df(u^{(k)})(d^{(k)}) \leq Df(\hat{u})(\hat{d}).$$

The proof of this lemma can be found in [212, p. 468].

Lemma 2.4.2 *It holds for every sequence* $(u^{(k)})_{k\in\mathbb{N}}$ *generated by* (2.43) *and for all* $u \in H$ *that*

$$D_\varphi^{(k+1)}(u, u^{(k+1)}) \leq D_\varphi^{(k)}(u, u^{(k)}) - D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}), \quad \textit{if} \quad \mathcal{F}(u) \leq \mathcal{F}(u^{(k+1)}), \tag{2.47}$$

$$\mathcal{F}(u^{(k+1)}) - \mathcal{F}(u) \leq \frac{1}{k\gamma} D_\varphi^{(1)}(u, u^{(1)}). \tag{2.48}$$

Proof: The following proof can also be found in [136, pp. 1148-1150]. A short calculation shows that by definition of $D_\varphi^{(k)}$ we have

$$D_\varphi^{(k+1)}(u, u^{(k+1)}) - D_\varphi^{(k)}(u, u^{(k)}) + D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) = \langle p^{(k)} - p^{(k+1)}, u - u^{(k+1)} \rangle. \tag{2.49}$$

Since both functions φ and $\mathcal{F}$ are proper, convex and l.s.c. we have $\operatorname{dom} D_\varphi^{(k)}(\cdot, u^{(k)}) = H$ and $0 \in \operatorname{int}(\operatorname{dom} \mathcal{F} - \operatorname{dom} D_\varphi^{(k)}(\cdot, u^{(k)}))$. Hence, it holds for $u^{(k+1)}$ and $p^{(k+1)}$ that

$$0 \in \frac{1}{\gamma} \partial_u D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) + \partial \mathcal{F}(u^{(k+1)}) = \frac{1}{\gamma} (\partial \varphi(u^{(k+1)}) - p^{(k)}) + \partial \mathcal{F}(u^{(k+1)}).$$

This yields together with $p^{(k+1)} \in \partial \varphi(u^{(k+1)})$ that

$$p^{(k)} - p^{(k+1)} \in \gamma \partial \mathcal{F}(u^{(k+1)}).$$

So, it follows from (2.49) and the definition of the subdifferential that

$$D_\varphi^{(k+1)}(u, u^{(k+1)}) - D_\varphi^{(k)}(u, u^{(k)}) + D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) \leq \gamma (\mathcal{F}(u) - \mathcal{F}(u^{(k+1)})) \tag{2.50}$$

and for all $u \in H$ with $\mathcal{F}(u) \leq \mathcal{F}(u^{(k+1)})$ we have shown (2.47).
To prove the second statement we first observe that by the definition of $u^{(k+1)}$ in (2.43) we have

$$\mathcal{F}(u^{(k+1)}) + \frac{1}{\gamma} D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) \leq \mathcal{F}(u^{(k)}) + \frac{1}{\gamma} \underbrace{D_\varphi^{(k)}(u^{(k)}, u^{(k)})}_{=0}.$$

If we multiply both sides by $(k-1)\gamma$ and sum from one to l we obtain

$$\sum_{k=1}^{l} \frac{(k-1)\gamma}{\gamma} D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) \leq \gamma \sum_{k=1}^{l} (k-1)(\mathcal{F}(u^{(k)}) - \mathcal{F}(u^{(k+1)})),$$

or equivalently

$$\sum_{k=1}^{l} (k-1) D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) \leq -l\gamma \mathcal{F}(u^{(l+1)}) + \gamma \sum_{k=1}^{l} \mathcal{F}(u^{(k+1)}). \tag{2.51}$$

If we sum up (2.50) from one to l we get

$$D_\varphi^{(l+1)}(u, u^{(l+1)}) + \sum_{k=1}^{l} D_\varphi^{(k)}(u^{(k+1)}, u^{(k)}) - D_\varphi^{(1)}(u, u^{(1)}) \leq l\gamma\mathcal{F}(u) - \gamma \sum_{k=1}^{l} \mathcal{F}(u^{(k+1)}). \quad (2.52)$$

Adding (2.51) and (2.52) yields

$$D_\varphi^{(l+1)}(u, u^{(l+1)}) - D_\varphi^{(1)}(u, u^{(1)}) + l\gamma(\mathcal{F}(u^{(l+1)}) - \mathcal{F}(u)) \leq -\sum_{k=1}^{l} kD_\varphi^{(k)}(u^{(k+1)}, u^{(k)}).$$

Since $D_\varphi^{(k)}(\cdot, u^{(k)}) \geq 0,\ \forall k \in \mathbb{N}$ we can drop the right-hand side and the first term on the left-hand side which gives us (2.48). □

Using the preceding lemmas we can prove the following convergence result.

Theorem 2.4.3 *Let $\mathcal{F}$ be a proper, convex and l.s.c. function which attains its minimum. Suppose an arbitrary initial value $u^{(0)}$, a parameter $\gamma > 0$ and a Bregman function which satisfies A1-A3 are given. Then, the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by the corresponding Bregman proximal point algorithm* (2.43) *converges to a minimizer of $\mathcal{F}$.*

Proof: This proof is adapted from [136] and [135] for our special setting, in particular, assumption A1. Since $\gamma > 0$ we obtain form (2.48) that

$$\lim_{k\to\infty} \mathcal{F}(u^{(k)}) = \min_{u\in H} \mathcal{F}(u)$$

which means that $\mathcal{F}(u^{(k)})$ is a minimizing sequence. Now let $\hat{u}$ be a minimizer of $\mathcal{F}$. Using again that $D_\varphi^{(k)}(\cdot, u^{(k)}) \geq 0$ for all k we can conclude from (2.47) that $D_\varphi^{(k)}(\hat{u}, u^{(k)}) \leq D_\varphi^{(1)}(\hat{u}, u^{(1)})$. We know by (2.46) that $D_\varphi'(\cdot, u^{(k)}) \leq D_\varphi^{(k)}(\cdot, u^{(k)})$ and thus it follows that

$$D_\varphi'(\hat{u}, u^{(k)}) \leq \alpha$$

with $\alpha := D_\varphi^{(1)}(\hat{u}, u^{(1)})$. In other words, we have

$$u^{(k)} \in \mathrm{lev}_\alpha D_\varphi'(\hat{u}, \cdot), \quad \forall k \in \mathbb{N},$$

where lev_α denotes the *level set* with respect to the parameter α. The facts that we are working in a finite-dimensional Hilbert space and that φ is *strictly* convex imply that the level sets $\mathrm{lev}_\alpha D_\varphi'(\hat{u}, \cdot)$ are bounded and thus $(u^{(k)})_{k\in\mathbb{N}}$ is bounded. Hence, there exists a subsequence $(u^{(k_l)})_{l\in\mathbb{N}}$ which converges to a minimizer $\tilde{u}$. We now show that

$$D_\varphi^\sharp(\tilde{u}, u^{(k_l)}) = \varphi(\tilde{u}) - \varphi(u^{(k_l)}) + \sup\{\langle w, u^{(k_l)} - \tilde{u}\rangle \ :\ w \in \partial\varphi(u^{(k_l)})\} \to 0.$$

Because of assumptions A1-A3, the function φ is continuous on H, see [181, Theorem 10.2], and thus $\varphi(u^{(k_l)}) \to \varphi(\tilde{u})$. It remains to show that

$$\sup\{\langle w, u^{(k_l)} - \tilde{u}\rangle \ :\ w \in \partial\varphi(u^{(k_l)})\} \to 0. \quad (2.53)$$

Suppose (2.53) does not hold true. Then, there must be an unbounded sequence $w^{(l)} \in \partial\varphi(u^{(k_l)})$. Since $(u^{(k_l)})_{l\in\mathbb{N}}$ is bounded, we can thus construct a bounded sequence $(y^{(l)})_{l\in\mathbb{N}}$ such that $\langle w^{(l)}, y^{(l)} - u^{(k_l)}\rangle \to +\infty$ by setting

$$y^{(l)} := u^{(k_l)} + \frac{w^{(l)}}{\|w^{(l)}\|}.$$

This yields a contradiction in the following way: Since $w^{(l)} \in \partial\varphi(u^{(k_l)})$ we have

$$\langle w^{(l)}, y - u^{(k_l)}\rangle \leq \varphi(y) - \varphi(u^{(k_l)}), \quad \forall y \in H.$$

For $y := y^{(l)}$ we thus have

$$\langle w^{(l)}, y^{(l)} - u^{(k_l)}\rangle \leq \varphi(y^{(l)}) - \varphi(u^{(k_l)}), \quad \forall l \in \mathbb{N}. \tag{2.54}$$

By assumption, H is finite-dimensional (A1) so that it is a locally compact space. Hence, φ being continuous implies that φ is locally bounded. It follows from the boundedness of $(y^{(l)})_{l\in\mathbb{N}}$ and $(u^{(k_l)})_{l\in\mathbb{N}}$ that the right-hand side of (2.54) is bounded. The left-hand side, however, goes to infinity which yields the contradiction.

So, we have that (2.53) holds true and thus $D_\varphi^\sharp(\tilde{u}, u^{(k_l)}) \to 0$. Consequently, by (2.46) we have

$$D_\varphi^{(k_l)}(\tilde{u}, u^{(k_l)}) \to 0.$$

If we apply (2.47) and the fact that $D_\varphi^{(k)}(\cdot, u^{(k)}) \geq 0$ we obtain

$$D_\varphi^{(k)}(\tilde{u}, u^{(k)}) \leq D_\varphi^{(k_l)}(\tilde{u}, u^{(k_l)}), \quad \forall k > k_l$$

and so we get $D_\varphi^{(k)}(\tilde{u}, u^{(k)}) \to 0$. By (2.46) this gives

$$D'_\varphi(\tilde{u}, u^{(k)}) \to 0. \tag{2.55}$$

We now prove that (2.55) yields $u^{(k)} \to \tilde{u}$. Assume that $u^{(k)}$ does not converge to $\tilde{u}$. Since the sequence $(u^{(k)})_{k\in\mathbb{N}}$ is bounded there exists a convergent subsequence $(u^{(k_m)})_{m\in\mathbb{N}}$ with limit $\tilde{u} + z$, $z \neq 0$. Recall that

$$D'_\varphi(\tilde{u}, u^{(k_m)}) = \varphi(\tilde{u}) - \varphi(u^{(k_m)}) - D\varphi(u^{(k_m)})(\tilde{u} - u^{(k_m)}).$$

By the continuity of φ, it follows that $\varphi(u^{(k_m)}) \to \varphi(\tilde{u} + z)$ and Lemma 2.4.1 implies that

$$\limsup_{k\to\infty} D\varphi(u^{(k_m)})(\tilde{u} - u^{(k_m)}) \leq D\varphi(\tilde{u} + z)(-z).$$

Consequently,

$$\varphi(\tilde{u}) - \varphi(\tilde{u} + z) - D\varphi(\tilde{u} + z)(-z) \leq 0.$$

However, by assumption A3, the function φ is strictly convex so that (compare [181, p. 214])

$$D\varphi(\tilde{u} + z)(-z) < \varphi(\tilde{u}) - \varphi(\tilde{u} + z)$$

which yields a contradiction. It follows that $z = 0$ or, in other words, $u^{(k)} \to \tilde{u}$. □

Let us return to our initial problems (P) and (D) defined in Section 2.2. The classical example of a Bregman function satisfying the properties A1-A3 is $\varphi := \frac{1}{2}\|\cdot\|^2$. The corresponding Bregman distance is given by

$$D^{(k)}_{\frac{1}{2}\|\cdot\|^2}(u, u^{(k)}) = \frac{1}{2}\|u - u^{(k)}\|^2. \tag{2.56}$$

This Bregman distance is symmetric which is not true for general Bregman distances. However, the triangle inequality does not hold true for this example either. Applying the BPP method (2.43) with (2.56), we recover the *classical proximal point* (PP) *algorithm* for (P) dating back to [155] which can be written as follows, compare [184],

$$u^{(k+1)} = \operatorname{prox}_{\gamma\mathcal{F}_P}(u^{(k)}) = \operatorname*{argmin}_{u\in H_1} \big\{\frac{1}{2\gamma}\|u - u^{(k)}\|^2 + \mathcal{F}_P(u)\big\} = J_{\gamma\partial\mathcal{F}_P}(u^{(k)}). \tag{2.57}$$

So, this proximal point algorithm is simply the iterated application of the resolvent of the full mapping $\gamma\partial\mathcal{F}_P$. Clearly, by Lemma 2.3.4 the resolvent is firmly nonexpansive for our choice of g, Φ and D, so that as an alternative to Theorem 2.4.3 we can conclude from Theorem 2.3.9 that the PP algorithm is guaranteed to converge for any initial point $u^{(0)}$ (weakly in general Hilbert spaces). Recall that the aim of the preceding sections was to develop splitting methods which allow us to compute a minimizer without having to evaluate the full resolvent of the subdifferential of $\mathcal{F}_P$ which in general cannot be done by a fast algorithm.

In the same way as in (2.57), we can define the proximal point algorithm for (D)

$$b^{(k+1)} = \operatorname{prox}_{\gamma\partial\mathcal{F}_D}(b^{(k)}) = \operatorname*{argmin}_{b\in H_2} \big\{\frac{1}{2\gamma}\|b - b^{(k)}\|^2 + \mathcal{F}_D(b)\big\} = J_{\gamma\partial\mathcal{F}_D}(b^{(k)}) \tag{2.58}$$

and the same convergence result holds true.

2.4.2 Split Bregman and augmented Lagrangian method

We will now apply the BPP algorithm to the following equivalent constrained version of the primal problem (P):

$$(P_{Split}) \qquad \min_{u\in H_1, d\in H_2} E(u, d) \quad \text{s.t.} \quad Du = d,$$

where

$$E(u, d) := g(u) + \Phi(d).$$

Note that in the context of total variation regularization this splitting idea was first used in [217] where the authors introduce a quadratic penalty term to incorporate the constraint. In the second part of this Section, we will show the relation of the resulting algorithm to the augmented Lagrangian and the PP method (2.58) applied to (D).

The split Bregman method. Following [120], we consider the BPP algorithm for the objective function $\mathcal{F}(u) := \frac{1}{2\gamma}\|Du - d\|^2$ with the Bregman distance

$$D_E^{(p^{(k)})}(u, d, u^{(k)}, d^{(k)}) = E(u, d) - E(u^{(k)}, d^{(k)}) - \langle p_u^{(k)}, u - u^{(k)}\rangle - \langle p_d^{(k)}, d - d^{(k)}\rangle,$$

with $\big(p_u^{(k)}, p_d^{(k)}\big) \in \partial E(u^{(k)}, d^{(k)})$. This results in the algorithm

$$\begin{aligned}(u^{(k+1)}, d^{(k+1)}) &= \operatorname*{argmin}_{u\in H_1, d\in H_2} \big\{D_E^{(p^{(k)})}(u, d, u^{(k)}, d^{(k)}) + \frac{1}{2\gamma}\|Du - d\|^2\big\}, \qquad (2.59)\\ p_u^{(k+1)} &= p_u^{(k)} - \frac{1}{\gamma}D^*(Du^{(k+1)} - d^{(k+1)}),\\ p_d^{(k+1)} &= p_d^{(k)} + \frac{1}{\gamma}(Du^{(k+1)} - d^{(k+1)}),\end{aligned}$$

where we have used that (2.59) implies

$$\begin{aligned}0 \;\in\; &\partial E(u^{(k+1)}, d^{(k+1)}) - \big(p_u^{(k)}, p_d^{(k)}\big) + \big(\frac{1}{\gamma}D^*(Du^{(k+1)} - d^{(k+1)}), -\frac{1}{\gamma}(Du^{(k+1)} - d^{(k+1)})\big),\\ &= \partial E(u^{(k+1)}, d^{(k+1)}) - \big(p_u^{(k+1)}, p_d^{(k+1)}\big),\end{aligned}$$

so that $\big(p_u^{(k)}, p_d^{(k)}\big) \in \partial E(u^{(k)}, d^{(k)})$. Setting $p_u^{(k)} = -\frac{1}{\gamma}D^*b^{(k)}$ and $p_d^{(k)} = \frac{1}{\gamma}b^{(k)}$ for all $k \geq 0$ and regarding that for a bounded linear operator D

$$\begin{aligned}&D_E^{(p^{(k)})}(u, d, u^{(k)}, d^{(k)}) + \frac{1}{2\gamma}\|Du - d\|^2\\ =\;& E(u, d) - E(u^{(k)}, d^{(k)}) + \frac{1}{\gamma}\langle b^{(k)}, Du - Du^{(k)}\rangle - \frac{1}{\gamma}\langle b^{(k)}, d - d^{(k)}\rangle + \frac{1}{2\gamma}\|Du - d\|^2,\end{aligned}$$

Goldstein and Osher obtained the *split Bregman method* in [120]:

Algorithm (Split Bregman algorithm)
Initialization: $u^{(0)}, p^{(0)}$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned}(u^{(k+1)}, d^{(k+1)}) &= \operatorname*{argmin}_{u\in H_1, d\in H_2} \big\{E(u, d) + \frac{1}{2\gamma}\|b^{(k)} + Du - d\|^2\big\},\\ b^{(k+1)} &= b^{(k)} + Du^{(k+1)} - d^{(k+1)}. \qquad (2.60)\end{aligned}$$

Unfortunately, Theorem 2.4.3 which establishes convergence for Bregman iterations can not be applied directly here since even if we restrict ourselves to the finite-dimensional setting the function Φ in E does not satisfy A3. However, we can show convergence of $(b^{(k)})_{k\in\mathbb{N}}$ by deriving the split Bregman algorithm in another way, namely as an augmented Lagrangian method described in the next paragraph. Moreover, we will see that the scaled

sequence $(\frac{1}{\gamma}b^{(k)})_{k\in\mathbb{N}}$ of the split Bregman algorithm converges to a solution of (D).

If the convergence of $(b^{(k)})_{k\in\mathbb{N}}$ is clear, it remains to examine the convergence properties of the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by (2.60). To this end, we assume now that H_1 and H_2 are finite-dimensional and that the sequence $(\frac{1}{\gamma}b^{(k)})_{k\in\mathbb{N}}$ converges to a solution of the dual problem. We obtain the following results.

Proposition 2.4.4 *Assume that H_1 and H_2 are finite-dimensional Hilbert spaces. Then, every cluster point of $(u^{(k)})_{k\in\mathbb{N}}$ generated by the split Bregman algorithm* (2.60) *is a solution of the primal problem (P).*

Proof: The split Bregman algorithm (2.60) can be written as

$$\begin{aligned} 0 &\in \partial g(u^{(k+1)}) + \frac{1}{\gamma}D^*(Du^{(k+1)} - d^{(k+1)} + b^{(k)}), && (2.61)\\ 0 &\in \partial\Phi(d^{(k+1)}) - \frac{1}{\gamma}(Du^{(k+1)} - d^{(k+1)} + b^{(k)}), && (2.62)\\ b^{(k+1)} &= b^{(k)} + Du^{(k+1)} - d^{(k+1)}. && (2.63) \end{aligned}$$

Let $\hat{b}$ and $\hat{d}$ denote the limits of the sequences $(b^{(k)})_{k\in\mathbb{N}}$ and $(d^{(k)})_{k\in\mathbb{N}}$, respectively. Furthermore, let $\hat{u}$ be a cluster point of $(u^{(k)})_{k\in\mathbb{N}}$ with respect to a convergent subsequence $(u^{(k_l)})_{l\in\mathbb{N}}$. Because the subdifferential of the functions g and Φ are maximal monotone we can pass to the limits with respect to the indices k_l in (2.61) and (2.62), see Theorem 2.2.2. This gives the Karush-Kuhn-Tucker conditions

$$\begin{aligned} &0 \in \partial g(\hat{u}) + \tfrac{1}{\gamma}D^*\hat{b},\\ &0 \in \partial\Phi(\hat{d}) - \tfrac{1}{\gamma}\hat{b}, \end{aligned} \tag{2.64}$$

or equivalently,

$$\begin{aligned} &0 \in \partial g(\hat{u}) + D^*(\tfrac{1}{\gamma}\hat{b}),\\ &0 \in \partial\Phi^*(\tfrac{1}{\gamma}\hat{b}) - D\hat{u} \end{aligned} \tag{2.65}$$

of problems (P) and (D), compare (2.8)-(2.9). Hence, $\hat{u}$ and $\frac{1}{\gamma}\hat{b}$ solve the primal problem and the dual problem, respectively. □

Theorem 2.4.5 *Suppose that H_1 and H_2 are finite-dimensional Hilbert spaces and that the primal problem (P) has a* unique *solution. Then, the sequence $(u^{(k)})_{k\in\mathbb{N}}$, generated by the split Bregman algorithm, converges to the solution of (P).*

Proof: The proof of this theorem employs the concept of coercivity of the objective function in the first step of (2.60). For detail, we refer to the proof of Corollary 2.4.10 in Section 2.4.3.

We will now characterize the split Bregman algorithm as an augmented Lagrangian algorithm. This fact was already noticed in [120, 206, 230].

The augmented Lagrangian method. We start by considering the convex, equality-constrained minimization problem of the form

$$\operatorname*{argmin}_{u\in H_1} h(u) \quad \text{s.t.} \quad \Theta(u) = 0, \tag{2.66}$$

where $h : H_1 \to \overline{\mathbb{R}}$ is a proper, convex and l.s.c. functions on a Hilbert space H_1 and

$$0 \in \operatorname{int}(\operatorname{dom} h - \operatorname{dom} \iota_{\Theta^{-1}(\{0\})}).$$

Later we will specialize to the problem (P_{Split}). For the functional $\Theta : H_1 \to H_2$, we consider two scenarios:

i) First, Θ is a bounded linear operator and H_1 and H_2 are general Hilbert spaces.

ii) In a second scenario, H_2 is a separable Hilbert space and we assume that for some fixed countable orthonormal basis the mapping

$$\Theta(u) = (\dots, \Theta_{-1}(u), \Theta_0(u), \Theta_1(u), \dots), \quad \forall u \in H_1,$$

has the property that the functions $\Theta_i : H \to \mathbb{R}$ are proper, convex and l.s.c.

For a discussion of the augmented Lagrangian method in more general spaces the reader is referred to [108]. In terms of the Lagrangian

$$L(u, p) = h(u) + \langle p, \Theta(u)\rangle,$$

the primal problem (2.66) can be written as

$$\operatorname*{argmin}_{u\in H_1}\{\sup_{p\in H_2} L(u, p)\}$$

and the corresponding dual problem has the form

$$\operatorname*{argmax}_{p\in H_2}\{\inf_{u\in H_1} L(u, p)\}. \tag{2.67}$$

The *augmented Lagrangian algorithm* (AL) which goes back to [128, 179, 182], applied to the primal problem (2.66), is defined as follows:

Algorithm (Augmented Lagrangian algorithm (AL))
Initialization: $u^{(0)}, p^{(0)}$
For $k = 0, 1, \dots$ repeat until a stopping criterion is reached

$$u^{(k+1)} = \operatorname*{argmin}_{u\in H_1}\{L(u, p^{(k)}) + \frac{\gamma}{2}\|\Theta(u)\|^2\}, \tag{2.68}$$

$$p^{(k+1)} = p^{(k)} + \gamma\Theta(u^{(k+1)}). \tag{2.69}$$

Note that the AL algorithm is also called the *method of multipliers*. The term augmented Lagrangian is motivated by the Lagrangian being augmented by the quadratic penalty term in (2.68), compare [169]. It is important here that the parameter γ is fixed in the augmented Lagrangian algorithm. If we enforce the equality constraint by a simple *quadratic penalty algorithm*

$$u^{(k+1)} = \operatorname*{argmin}_{u \in H_1} \{h(u) + \frac{\gamma_k}{2} \|\Theta(u)\|^2\},$$

the parameter γ_k has to go to infinity which can cause numerical instability.

It is well-known that the AL algorithm is equivalent to the PP algorithm applied to the dual problem (D) which is given by

$$p^{(k+1)} = \operatorname*{argmin}_{p \in H_2} \{\frac{1}{2\gamma} \|p - p^{(k)}\|^2 - \inf_{u \in H_1} L(u, p)\}. \tag{2.70}$$

The following theorem shows that the sequence $(p^{(k)})_{k \in \mathbb{N}}$ generated by the PP algorithm (2.70) coincides with the one produced by the AL algorithm (2.68)-(2.69). Thus, by the results of Section 2.4.2 we can conclude that the sequence $(p^{(k)})_{k \in \mathbb{N}}$ of the AL algorithm converges weakly.

Theorem 2.4.6 *The sequence $(p^{(k)})_{k \in \mathbb{N}}$ generated by* (2.70) *coincides with the one computed by* (2.68)-(2.69) *for the same starting point $p^{(0)}$.*

Proof: For this proof we follow the lines of [46, Theorem 3.4.7]. Let $(\tilde{p}^{(k)})_{k \in \mathbb{N}}$ denote the sequence generated by (2.70) and let $(u^{(k)}, p^{(k)})_{k \in \mathbb{N}}$ be the solutions of (2.68)-(2.69) in each iteration. We start with $\tilde{p}^{(0)} = p^{(0)}$ and assume that $\tilde{p}^{(k)} = p^{(k)}$ for some $k \in \mathbb{N}$. If we can show that $p^{(k+1)}$ also solves (2.70) then we are done because the objective function of (2.70) is strictly convex and coercive so that the solution exists and is unique. Hence, we have to prove that

$$0 \in \partial\big(\frac{1}{2\gamma} \|\cdot - \tilde{p}^{(k)}\|^2 \underbrace{- \inf_{u \in H_1} L(u, \cdot)}_{:=\phi(\cdot)}\big)(p^{(k+1)}). \tag{2.71}$$

To this end, we first show that for a pair $(\bar{u}, \bar{p})$ with the property that $\bar{u}$ minimizes $L(\cdot, \bar{p})$ we have

$$-\Theta(\bar{u}) \in \partial\phi(\bar{p}). \tag{2.72}$$

Using the definition of the subdifferential we can write (2.72) in the equivalent form

$$\langle -\Theta(\bar{u}), p - \bar{p} \rangle \leq \phi(p) - \phi(\bar{p}), \quad \forall p \in H_2. \tag{2.73}$$

To show (2.73), we use the definition of L and obtain

$$\begin{aligned} & \phi(p) + L(\bar{u}, \bar{p}) - \langle -\Theta(\bar{u}), p - \bar{p} \rangle \\ = \; & \phi(p) + h(\bar{u}) + \langle \Theta(\bar{u}), p \rangle \\ = \; & \underbrace{\phi(p)}_{-\inf_u L(u,p)} + L(\bar{u}, p) \geq 0, \quad \forall p \in H_2, \end{aligned}$$

which is the same as (2.73) since $\phi(\bar{p}) = -L(\bar{u}, \bar{p})$.

In our situation, it indeed holds true that $u^{(k+1)}$ minimizes $L(\cdot, p^{(k+1)})$: In the first scenario, where Θ is a bounded linear operator, we have for (2.68)

$$\begin{aligned} & u^{(k+1)} = \operatorname*{argmin}_{u \in H_1} \{L(u, p^{(k)}) + \frac{\gamma}{2}\|\Theta(u)\|^2\} \\ \Leftrightarrow \quad & 0 \in \partial h(u^{(k+1)}) + \Theta^*(p^{(k)}) + \gamma \Theta^* \Theta u^{(k+1)}. \end{aligned}$$

Using (2.69), we thus obtain

$$\begin{aligned} & 0 \in \partial h(u^{(k+1)}) + \Theta^*(p^{(k+1)} - \gamma \Theta u^{(k+1)}) + \gamma \Theta^* \Theta u^{(k+1)} \\ \Leftrightarrow \quad & 0 \in \partial h(u^{(k+1)}) + \Theta^* p^{(k+1)} \\ \Leftrightarrow \quad & u^{(k+1)} = \operatorname*{argmin}_{u \in H_1} L(u, p^{(k+1)}). \end{aligned}$$

For the second scenario, where H_2 is separable, it holds that

$$\begin{aligned} & u^{(k+1)} = \operatorname*{argmin}_{u \in H_1} \{L(u, p^{(k)}) + \frac{\gamma}{2}\|\Theta(u)\|^2\} \\ \Leftrightarrow \quad & 0 \in \partial h(u^{(k+1)}) + \sum_{i=-\infty}^{\infty} p_i^{(k)} \partial \Theta_i(u^{(k+1)}) + \gamma \sum_{i=-\infty}^{\infty} \Theta_i(u^{(k+1)}) \partial \Theta_i(u^{(k+1)}) \end{aligned}$$

and again we get from (2.69) that

$$\begin{aligned} & 0 \in \partial h(u^{(k+1)}) + \sum_{i=-\infty}^{\infty} (p_i^{(k+1)} - \gamma \Theta_i u^{(k+1)}) \partial \Theta_i(u^{(k+1)}) + \gamma \sum_{i=-\infty}^{\infty} \Theta_i(u^{(k+1)}) \partial \Theta_i(u^{(k+1)}) \\ \Leftrightarrow \quad & u^{(k+1)} = \operatorname*{argmin}_{u \in H_1} L(u, p^{(k+1)}). \end{aligned}$$

Now we can apply

$$-\Theta(u^{(k+1)}) \in \partial \phi(p^{(k+1)})$$

to (2.71). Note that the domain of the quadratic term is the whole space H_2 so that we can subdifferentiate the two terms in (2.71) separately. Hence, we obtain

$$\frac{1}{\gamma}(p^{(k+1)} - \tilde{p}^{(k)}) - \Theta(u^{(k+1)}) \in \partial\big(\frac{1}{2\gamma}\|\cdot - \tilde{p}^{(k)}\|^2 - \inf_{u \in H} L(u, \cdot)\big)(p^{(k+1)}). \tag{2.74}$$

Using (2.69) and the induction hypothesis $\tilde{p}^{(k)} = p^{(k)}$, we see that the left-hand side of (2.74) is indeed the zero element. So, (2.71) is satisfied, i.e., $p^{(k+1)}$ is also the minimizer of (2.70) which concludes the proof. □

It is not surprising that the augmented Lagrangian method can also be applied to inequality constraint problems since in this case, too, it coincides with the PP method applied to the corresponding dual problem compare, e.g., [27, 132, 169, 184].

Observe that our problem (P_{Split}) is of the form (2.66) and thus we can apply the corresponding AL algorithm which is given by

$$\begin{aligned}(u^{(k+1)}, d^{(k+1)}) &= \operatorname*{argmin}_{u\in H_1, d\in H_2} \big\{E(u,d) + \langle b^{(k)}, Du - d\rangle + \frac{1}{2\gamma}\|Du-d\|^2\big\},\\ b^{(k+1)} &= b^{(k)} + \frac{1}{\gamma}(Du^{(k+1)} - d^{(k+1)}).\end{aligned} \tag{2.75}$$

We see that after rescaling $b^{(k)}$ by the factor $\frac{1}{\gamma}$ in (2.75), we obtain the split Bregman algorithm (2.60). By the identification of the AL algorithm with the PP algorithm for the dual problem via Theorem 2.4.6, we can immediately conclude that $(\frac{1}{\gamma}b^{(k)})_{k\in\mathbb{N}}$ generated by the split Bregman algorithm (2.75) converges to the solution of the dual problem (D).

Interestingly, we have seen three different points of view on the above split Bregman algorithm: an operator splitting, a Bregman and primal-dual Lagrangian approach. Let us summarize this in the following diagram:

PP for (D)	=	Split Bregman Algorithm	=	AL for (P)

2.4.3 Alternating split Bregman algorithm

In the first step of the split Bregman algorithm (2.60) we have to minimize over u and d simultaneously which is too hard to compute in many applications. Goldstein and Osher therefore proposed in [120], see also [119], to minimize with respect to these two variables alternatingly. This gives rise to the following *alternating split Bregman algorithm* to solve (P), resp. (P_{Split}):

Algorithm (Alternating split Bregman algorithm)
Initialization: $b^{(0)}, d^{(0)}$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$u^{(k+1)} = \operatorname*{argmin}_{u\in H_1} \big\{g(u) + \frac{1}{2\gamma}\|b^{(k)} + Du - d^{(k)}\|^2\big\}, \tag{2.76}$$

$$d^{(k+1)} = \operatorname*{argmin}_{d\in H_2} \big\{\Phi(d) + \frac{1}{2\gamma}\|b^{(k)} + Du^{(k+1)} - d\|^2\big\}, \tag{2.77}$$

$$b^{(k+1)} = b^{(k)} + Du^{(k+1)} - d^{(k+1)}. \tag{2.78}$$

Interestingly, the alternating split Bregman algorithm turns out to be equivalent to the Douglas-Rachford splitting algorithm studied in Section 2.3.3. Based on [94, 96, 109] and [116, pp. 95] we have the following important result, cf., [193, 192].

Theorem 2.4.7 *For any starting values $b^{(0)} \in H_2$ and $d^{(0)} \in H_2$ the alternating split Bregman algorithm is equal to the Douglas-Rachford splitting algorithm*

$$\begin{aligned}t^{(k+1)} &= J_{\eta A}(2p^{(k)} - t^{(k)}) + t^{(k)} - p^{(k)},\\ p^{(k+1)} &= J_{\eta B}(t^{(k+1)})\end{aligned}$$

applied to (D) with $A := \partial(g^* \circ (-D^*))$, $B := \partial\Phi^*$, *step length* $\eta = 1/\gamma$ *and initial values* $t^{(0)} = \eta(b^{(0)} + d^{(0)})$ *and* $p^{(0)} = \eta b^{(0)}$ *in the sense that*

$$\begin{aligned} t^{(k)} &= \eta(b^{(k)} + d^{(k)}), \\ p^{(k)} &= \eta b^{(k)}, \end{aligned} \tag{2.79}$$

for all $k > 0$.

Proof: 1. First, we show that for a proper, convex and l.s.c. function $h : H_1 \to \mathbb{R} \cup \{+\infty\}$ and a bounded linear operator $K : H_1 \to H_2$ the following relation holds true:

$$\hat{p} = \operatorname*{argmin}_{p \in H_1} \left\{ \frac{\eta}{2} \|Kp - q\|_2^2 + h(p) \right\} \quad \Rightarrow \quad \eta(K\hat{p} - q) = J_{\eta\,\partial(h^* \circ (-K^*))}(-\eta q). \tag{2.80}$$

The left-hand side of (2.80) is equivalent to

$$0 \in \eta K^*(K\hat{p} - q) + \partial h(\hat{p}) \quad \Leftrightarrow \quad \hat{p} \in \partial h^*\big(- \eta K^*(K\hat{p} - q)\big).$$

Applying $-\eta K$ on both sides and adding $-\eta q$ implies using the chain rule that

$$-\eta K\hat{p} \in -\eta K \partial h^*\big(- \eta K^*(K\hat{p} - q)\big) \;=\; \eta\, \partial\big(h^* \circ (-K^*)\big)\big(\eta(K\hat{p} - q)\big)$$

and thus we have

$$-\eta q \in \big(I + \eta\, \partial(h^* \circ (-K^*))\big)\big(\eta(K\hat{p} - q)\big),$$

which is by the definition of the resolvent equivalent to the right-hand side of (2.80).
2. Applying (2.80) to (2.76) with $h := g$, $K := D$ and $q := d^{(k)} - b^{(k)}$ yields

$$\eta(b^{(k)} + Du^{(k+1)} - d^{(k)}) = J_{\eta A}(\eta(b^{(k)} - d^{(k)})).$$

Assume that the alternating split Bregman iterates and the DRS iterates coincide with the identification (2.79) up to some $k \in \mathbb{N}$. Using this induction hypothesis it follows that

$$\eta(b^{(k)} + Du^{(k+1)}) = J_{\eta A}(\underbrace{\eta(b^{(k)} - d^{(k)})}_{2p^{(k)} - t^{(k)}}) + \underbrace{\eta d^{(k)}}_{t^{(k)} - p^{(k)}} = t^{(k+1)}. \tag{2.81}$$

By definition of $b^{(k+1)}$ in (2.78) we see that $\eta(b^{(k+1)} + d^{(k+1)}) = t^{(k+1)}$. Next we apply (2.80) to (2.77) with $h := \Phi$, $K := -I$ and $q := -b^{(k)} - Du^{(k+1)}$. Together with (2.81) this gives

$$\eta(-d^{(k+1)} + b^{(k)} + Du^{(k+1)}) = J_{\eta(\partial\Phi^*)}(\underbrace{\eta(b^{(k)} + Du^{(k+1)})}_{t^{(k+1)}}) = p^{(k+1)}. \tag{2.82}$$

Applying (2.78) to the right-hand side of (2.82) we obtain $\eta b^{(k+1)} = \frac{1}{\gamma} b^{(k+1)} = p^{(k+1)}$ which completes the proof. □

Recently, a different proof of convergence of the alternating split Bregman algorithm was proposed in [52]. Since in our case D is always a bounded linear operator, it was noted in [101] that the alternating split Bregman algorithm, in a slightly different form based on the AL algorithm, was already known in optimization theory under the name of *alternating direction method of multipliers*. The term alternating direction method of multipliers was coined by [109] but the algorithm itself can be traced back to [110, 117]. We will not explain this algorithm in more detail in this paper.

In the following diagram, we summarize the important equivalences discussed in this section.

Alternating Split Bregman Algorithm	=	Douglas-Rachford Splitting Algorithm for (D)	=	Alternating Direction Method of Multipliers

We again notice the three different points of view on this algorithm: We can derive it via an operator splitting, a Bregman method or a special primal-dual Lagrangian method.

It must be noted here that recently there is great interest in advanced methods for optimization problems in image processing. The alternating split Bregman/alternating direction method of multipliers is just one example of techniques which make use of the Lagrangian of the problem, i.e., both dual and primal variable, splitting ideas and the proximal point operations. We can only briefly mention here the primal-dual hybrid gradient descent method of [234, 235], the algorithm of Chen and Teboulle [70] and, as pointed out in [101], the alternating minimization method of [211].

It remains to consider the convergence properties of the primal sequence $(u^{(k)})_{k\in\mathbb{N}}$ of the alternating split Bregman algorithm. Let us now suppose that H_1 and H_2 are finite-dimensional. So, Theorem 2.4.7 provides us with a convergence result for the sequence $(b^{(k)})_{k\in\mathbb{N}}$ of the alternating split Bregman algorithm. More specifically, let $\hat{b}$ the limit of this sequence then $\frac{1}{\gamma}\hat{b}$ is a solution of (D). The following results hold true for the sequence $(u^{(k)})_{k\in\mathbb{N}}$.

Proposition 2.4.8 *Assume that H_1 and H_2 are finite-dimensional Hilbert spaces. Then, every cluster point of $(u^{(k)})_{k\in\mathbb{N}}$ generated by the alternating split Bregman algorithm is a solution of the primal problem (P).*

Proof: We rewrite (2.76)-(2.78) in the equivalent form

$$0 \in \partial g(u^{(k+1)}) + \frac{1}{\gamma}D^*b^{(k)} + \frac{1}{\gamma}D^*(Du^{(k+1)} - d^{(k)}), \tag{2.83}$$

$$0 \in \partial\Phi(d^{(k+1)}) - \frac{1}{\gamma}b^{(k)} - \frac{1}{\gamma}(Du^{(k+1)} - d^{(k+1)}), \tag{2.84}$$

$$b^{(k+1)} = b^{(k)} + Du^{(k+1)} - d^{(k+1)}. \tag{2.85}$$

Let $\hat{b}$ and $\hat{d}$ denote the limits of the sequences $(b^{(k)})_{k\in\mathbb{N}}$ and $(d^{(k)})_{k\in\mathbb{N}}$, respectively. Furthermore, let $\hat{u}$ be a cluster point of $(u^{(k)})_{k\in\mathbb{N}}$ with convergent subsequence $(u^{(k_l)})_{l\in\mathbb{N}}$. Because

the subdifferential of the functions g and Φ are maximal monotone we can use Theorem 2.2.2 and pass to the limits with respect to the indices k_l in (2.83) and (2.84). This gives the Karush-Kuhn-Tucker conditions of problems (P) and (D)

$$\begin{aligned} 0 &\in \partial g(\hat{u}) + \tfrac{1}{\gamma} D^* \hat{b}, \\ 0 &\in \partial \Phi(\hat{d}) - \tfrac{1}{\gamma} \hat{b}, \end{aligned} \tag{2.86}$$

or equivalently,

$$\begin{aligned} 0 &\in \partial g(\hat{u}) + D^*(\tfrac{1}{\gamma} \hat{b}), \\ 0 &\in \partial \Phi^*(\tfrac{1}{\gamma} \hat{b}) - D\hat{u}. \end{aligned} \tag{2.87}$$

Thus, $\hat{u}$ and $\hat{b}$ solve the primal problem and the dual problem, respectively. □

Theorem 2.4.9 *Assume that H_1 and H_2 are finite-dimensional Hilbert spaces and let $\hat{b}$ and $\hat{d}$ be the limit points arising from the alternating split Bregman algorithm* (2.76)-(2.78)*. Then, a sufficient condition for $(u^{(k)})_{k\in\mathbb{N}}$ to converge to a solution of (P) is that*

$$\operatorname*{argmin}_{u\in H_1} \left\{ g(u) + \frac{1}{2\gamma} \|\hat{b} + Du - \hat{d}\|^2 \right\} \tag{2.88}$$

has a unique *solution.*

Proof: Let us rewrite (2.83) as

$$u^{(k+1)} \in (D^*D + \gamma \partial g)^{-1}(D^*(d^{(k)} - b^{(k)})).$$

The mapping $(D^*D + \gamma\partial g)^{-1}$ is maximal monotone because $D^*D + \gamma\partial g$ is maximal monotone as a subdifferential of a proper, l.s.c. and convex function and the inverse operator of every maximal monotone operator is maximal monotone. Hence, $(D^*D + \gamma\partial g)^{-1}$ being single-valued at the point $D^*(\hat{d} - \hat{b})$ implies in our finite-dimensional setting that it is continuous at this point, see, [185, Theorem 12.63 (c)]. Since $D^*(d^{(k)} - b^{(k)}) \to D^*(\hat{d} - \hat{b})$ as $k \to \infty$, it follows from the definition of continuity of set-valued mappings, cf. [185, Chapter 4], that the sequence $(u^{(k)})_{k\in\mathbb{N}}$ converges to the unique solution of (2.88). By Proposition 2.4.8, this element must be a solution of (P). □

In many examples the following special case of Theorem 2.4.9 can be used.

Corollary 2.4.10 *If H_1 and H_2 are finite-dimensional Hilbert spaces and if the primal problem (P) has a* unique *solution then $(u^{(k)})_{k\in\mathbb{N}}$, defined by the alternating split Bregman algorithm* (2.76)-(2.78)*, converges to the solution of (P).*

To prove the above corollary we will make use of the following lemma.

Lemma 2.4.11 *Let F be a l.s.c. and convex function on a finite-dimensional Hilbert space H. Assume that F has a unique minimizer, then it is coercive.*

Proof: Suppose that F is not coercive, i.e., there exists a sequence $(u^{(k)})_{k\in\mathbb{N}}$ with $\|u^{(k)}\| \to \infty$ as $k \to \infty$ and $|F(u^{(k)})| \leq C < +\infty$ for all $k \in \mathbb{N}$ and a constant C. Without loss of generality, assume that $\hat{u} = 0$ is the unique minimizer of F and that $F(0) = 0$ (otherwise use similar arguments as in the proof of Corollary 2.4.10 below). We consider the sequence defined by

$$v^{(k)} = \frac{u^{(k)}}{\|u^{(k)}\|},$$

which is clearly bounded and thus has a cluster point $\hat{v}$. The convexity of F yields for k large enough that

$$F(v^{(k)}) \leq \frac{1}{\|u^{(k)}\|} F(u^{(k)}) \to 0 \quad \text{for } k \to \infty$$

Since F is l.s.c. we obtain $F(\hat{v}) \leq 0$ and thus $F(\hat{v}) = 0$. This contradicts the uniqueness of the minimizer because by construction $\|\hat{v}\| = 1$. □

Proof of Corollary 2.4.10: It is sufficient to show that $(u^{(k)})_{k\in\mathbb{N}}$ is bounded because we know from Proposition 2.4.8 that every cluster point is equivalent to the unique solution of (P). Taking (2.76) into account, we see that boundedness holds true if the functional $F : H_1 \to \mathbb{R} \cup \{+\infty\}$ defined by

$$F(u) = g(u) + \tfrac{1}{2}\|Du\|^2 \tag{2.89}$$

is coercive. From Lemma 2.4.11 we know that $g + \Phi \circ D$ is coercive. It remains to show that this implies the coercivity of F. Note that each element $u \in H_1$ has an orthogonal decomposition $u = u_1 + u_2$ with $u_1 \in \mathcal{N}(D)$, where $\mathcal{N}(D)$ is the null space of D. If $\mathcal{N}(D) = \{0\}$ we are done since g is convex and thus F is coercive. So, let $\mathcal{N}(D) \neq \{0\}$. If for a sequence $(u^{(k)})_{k\in\mathbb{N}}$ it holds that $\|u_2^{(k)}\| \to +\infty$ as $k \to +\infty$ then $(\|u^{(k)}\|)_{k\in\mathbb{N}}$ is unbounded. By convexity of g, we have to show that there cannot be an unbounded sequence $(u^{(k)})_{k\in\mathbb{N}}$ with

$$\|u_2^{(k)}\| \leq C_1 < +\infty, \quad \forall\, k \in \mathbb{N} \tag{2.90}$$

and $|g(u^{(k)})| \leq C_2 < +\infty$. Assume that such a sequence exists. For $v_0 \in H_1$ with $F(v_0) < +\infty$, we define for any $m \in \mathbb{N}$

$$v_m^{(k)} = v_0 + m\frac{u^{(k)} - v_0}{\|u^{(k)} - v_0\|}.$$

Then, $(v_m^{(k)})_{k\in\mathbb{N}}$ is a bounded sequence and thus has a cluster point v_m. Assumption (2.90) implies that $v_m \in v_0 + \mathcal{N}(D)$. The convexity of g yields for k large enough

$$\begin{aligned} g(v_m^{(k)}) &\leq \big(1 - \tfrac{m}{\|u^{(k)}-v_0\|}\big) g(v_0) + \tfrac{m}{\|u^{(k)}-v_0\|} g(u^{(k)}) \\ &\leq \big(\underbrace{1 - \tfrac{m}{\|u^{(k)}-v_0\|}}_{\to 1}\big) g(v_0) + \underbrace{\tfrac{m}{\|u^{(k)}-v_0\|}}_{\to 0} C_2. \end{aligned}$$

Since g is l.s.c. we obtain $g(v_m) \leq g(v_0)$. The sequence $(g(v_m))_{m\in\mathbb{N}}$ must be bounded from below because a solution of (P) exists. Hence, we have constructed an unbounded sequence $(v_m)_{m\in\mathbb{N}}$ for which the values of both g and $\Phi \circ D$ are bounded. This yields a contradiction since $g + \Phi \circ D$ is coercive. □

Observe that we get a similar result as in Theorem 2.4.7 if we change the roles of the operators A and B in the Douglas-Rachford algorithm. To this end let us change the order of the first two steps (2.76) and (2.77) in the alternating split Bregman algorithm. This gives rise to the algorithm

$$d^{(k+1)} = \operatorname*{argmin}_{d\in H_2} \big\{\Phi(d) + \frac{1}{2\gamma}\|b^{(k)} + Du^{(k)} - d\|_2^2\big\}, \tag{2.91}$$

$$u^{(k+1)} = \operatorname*{argmin}_{u\in H_1} \big\{g(u) + \frac{1}{2\gamma}\|b^{(k)} + Du - d^{(k+1)}\|_2^2\big\}, \tag{2.92}$$

$$b^{(k+1)} = b^{(k)} + Du^{(k+1)} - d^{(k+1)}. \tag{2.93}$$

Hence, we obtain the following result:

Corollary 2.4.12 *For any starting values $b^{(0)} \in H_2$ and $u^{(0)} \in H_1$ the version (2.91)-(2.93) of the alternating split Bregman algorithm coincides with the Douglas-Rachford splitting algorithm applied to (D) with $A := \partial\Phi^*$, $B := \partial(g^* \circ (-D^*))$, step length $\eta = 1/\gamma$, initial values $t^{(0)} := \eta(b^{(0)} + Du^{(0)})$ and $p^{(k)} := \eta b^{(k)}$ we have*

$$\begin{aligned} t^{(k)} &= \eta(b^{(k)} + Du^{(k)}), \\ p^{(k)} &= \eta b^{(k)}, \end{aligned}$$

for all $k > 0$.

Proof: See the proof of Theorem 2.4.7.

2.4.4 Multiple splittings

Next, we want to apply the splitting idea even further. Observe that in step (2.76) of the alternating split Bregman algorithm the operator D appears in front of the variable u in the quadratic term. This might lead to a rather complicated solution of the minimization problem for which, e.g., a separate iterative algorithm may be needed in each iteration. In this case, the problem of choosing an appropriate stopping criterion for the inner loops arises. Furthermore, a convergence result for the alternating split Bregman algorithm with inner loops only exists if these inner steps are computed to an increasingly high precision, cf. [96]. Hence, the question is if the splitting idea can be applied also to decouple the minimization problem in step (2.76). This is indeed possible by rewriting the constrained version (P_{Split}) of the primal problem in the following form, cf. [101],

$$\min_{u,v\in H_1, d\in H_2} \{\langle 0, u\rangle + g(v) + \Phi(d)\} \quad \text{s.t.} \quad \begin{pmatrix} I \\ D \end{pmatrix} u = \begin{pmatrix} v \\ d \end{pmatrix}. \tag{2.94}$$

It is easy to see that the corresponding split Bregman (or augmented Lagrangian) algorithm is exactly the same as (2.60). However, applying the alternating split Bregman algorithm to (2.94) with respect to the functionals $(u,v,d) \mapsto \langle 0,u\rangle$ and $(u,v,d) \mapsto g(v)+\Phi(d)$ yields

$$u^{(k+1)} = \operatorname*{argmin}_{u\in H_1} \big\{ \langle 0,u\rangle + \frac{1}{2\gamma} \| \begin{pmatrix} b_1^{(k)} \\ b_2^{(k)} \end{pmatrix} + \begin{pmatrix} I \\ D \end{pmatrix} u - \begin{pmatrix} v^{(k)} \\ d^{(k)} \end{pmatrix} \|^2 \big\}, \tag{2.95}$$

$$\begin{pmatrix} v^{(k+1)} \\ d^{(k+1)} \end{pmatrix} = \operatorname*{argmin}_{v\in H_1, d\in H_2} \big\{ g(v) + \Phi(d) + \frac{1}{2\gamma} \| \begin{pmatrix} b_1^{(k)} \\ b_2^{(k)} \end{pmatrix} + \begin{pmatrix} I \\ D \end{pmatrix} u^{(k+1)} - \begin{pmatrix} v \\ d \end{pmatrix} \|^2 \big\}, \tag{2.96}$$

$$b^{(k+1)} = b^{(k)} + \begin{pmatrix} I \\ D \end{pmatrix} u^{(k+1)} - \begin{pmatrix} v^{(k+1)} \\ d^{(k+1)} \end{pmatrix}. \tag{2.97}$$

Observe that the minimization problem in the second step (2.96) decouples automatically and we obtain

$$u^{(k+1)} = \operatorname*{argmin}_{u\in H_1} \big\{ \frac{1}{2} \| b_1^{(k)} + u - v^{(k)} \|^2 + \frac{1}{2} \| b_2^{(k)} + Du - d^{(k)} \|^2 \big\}, \tag{2.98}$$

$$v^{(k+1)} = \operatorname*{argmin}_{v\in H_1} \big\{ g(v) + \frac{1}{2\gamma} \| b_1^{(k)} + u^{(k+1)} - v \|^2 \big\}, \tag{2.99}$$

$$d^{(k+1)} = \operatorname*{argmin}_{d\in H_2} \big\{ \Phi(d) + \frac{1}{2\gamma} \| b_2^{(k)} + Du^{(k+1)} - d \|^2 \big\}, \tag{2.100}$$

$$b^{(k+1)} = b^{(k)} + \begin{pmatrix} I \\ D \end{pmatrix} u^{(k+1)} - \begin{pmatrix} v^{(k+1)} \\ d^{(k+1)} \end{pmatrix}. \tag{2.101}$$

In the first step (2.98) of the above algorithm, we have to invert the bounded linear operator $I + D^*D$. For our applications, we will address this issue in Chapter 3. The construction idea of algorithm (2.98)-(2.101) is also useful for more general problems with more than two terms in the objective function. So, let us consider the primal problem

$$\min_{u\in H} \sum_{i=1}^{m} g_i(D_i u), \tag{2.102}$$

where $D_i : H \to H_i$, $i = 1,\dots,m$ are bounded linear operators, $H, H_1, \dots, H_m$ are Hilbert spaces and the functionals $g_i : H_i \to \overline{\mathbb{R}}$ are assumed to be proper, convex and l.s.c. Setting $g := \langle 0, \cdot \rangle$, $D = (D_1, \dots, D_m)$ and $\Phi(v) := \sum_{i=1}^m g_i(v_i)$, $v \in H_1 \times \cdots \times H_m$, we see that (2.102) has the same structure as (P). We again assume here that the regularity conditions (2.11)-(2.12) are satisfied. In the same way as for (2.98)-(2.101) we obtain from

$$\min_{u\in H, v_1\in H_1,\dots,v_m\in H_m} \{ \langle 0,u\rangle + \sum_{i=1}^{m} g_i(D_i u) \} \quad \text{s.t.} \quad \begin{pmatrix} D_1 \\ \vdots \\ D_m \end{pmatrix} u = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}$$

the following alternating split Bregman algorithm, cf., [101]

$$
\begin{aligned}
u^{(k+1)} &= \operatorname*{argmin}_{u\in H} \{\frac{1}{2}\sum_{i=1}^{m} \|b_i^{(k)} + D_i u - v_i^{(k)}\|^2\}, && (2.103)\\
v_1^{(k+1)} &= \operatorname*{argmin}_{v_1\in H_1} \{g_1(v_1) + \frac{1}{2\gamma}\|b_1^{(k)} + D_1 u^{(k+1)} - v_1\|^2\}, && (2.104)\\
&\vdots \\
v_m^{(k+1)} &= \operatorname*{argmin}_{v_m\in H_m} \{g_m(v_m) + \frac{1}{2\gamma}\|b_m^{(k)} + D_m u^{(k+1)} - v_m\|^2\}, && (2.105)\\
b^{(k+1)} &= b^{(k)} + \begin{pmatrix} D_1 \\ \vdots \\ D_m \end{pmatrix} u^{(k+1)} - \begin{pmatrix} v_1^{(k+1)} \\ \vdots \\ v_m^{(k+1)} \end{pmatrix}. && (2.106)
\end{aligned}
$$

Note that again the minimization with respect to $v_1, \ldots, v_m$ is decoupled. In the first step, we have to invert the operator $\sum_{i=1}^{m} D_i^* D_i$.

We will now see that for the alternating split Bregman algorithm (2.103)-(2.106) we can establish a relation to the Douglas-Rachford splitting method as in the preceding subsection. This will also clarify the convergence properties.

We can derive the following result analogously to Theorem 2.4.7.

Theorem 2.4.13 *For any starting values $b_i^{(0)} \in H_i$, $v_i^{(0)} \in H_i$, $i = 1, \ldots, m$, the alternating split Bregman algorithm* (2.103)-(2.106) *is equivalent to the Douglas-Rachford splitting algorithm*

$$
\begin{aligned}
t^{(k+1)} &= J_{\eta A}(2p^{(k)} - t^{(k)}) + t^{(k)} - p^{(k)},\\
p^{(k+1)} &= J_{\eta B}(t^{(k+1)}),
\end{aligned}
$$

with respect to $A := \partial(\langle 0, \cdot\rangle \circ (-(D_1^* \cdots D_m^*)))$, B defined by
$B(p) := (\partial g_1^*(p_1), \ldots, \partial g_m^*(p_m))$, step length $\eta = 1/\gamma$ and initial values $t^{(0)} = \eta(b^{(0)} + v^{(0)})$ and $p^{(0)} = \eta b^{(0)}$. In particular, we have

$$
\begin{aligned}
t^{(k)} &= \eta(b^{(k)} + v^{(k)}),\\
p^{(k)} &= \eta b^{(k)},
\end{aligned}
$$

for all $k > 0$.

Proof: If we compare the alternating split Bregman algorithms (2.76)-(2.78) and (2.103)-(2.106) we see that the only difference is that the function g in the first algorithm now corresponds to the mapping $u \mapsto \langle 0, u\rangle$, Φ corresponds to $(v_1, \ldots, v_m) \mapsto \sum_{i=1}^{m} g_i(v_i)$ and the bounded linear operator D is now $(D_1, \ldots, D_m)$. So, as shown in Theorem 2.4.7, (2.103)-(2.106) is equivalent to the Douglas-Rachford splitting algorithm applied to the

operators A and B defined as above. $\square$

Let us now have a closer look at the common zero problem in the above Theorem 2.4.13 of finding an element $p \in H_1 \times \cdots \times H_m$ such that $0 \in A(p) + B(p)$. It holds for $x \in H$ that

$$\langle 0, \cdot \rangle^*(x) = \begin{cases} 0 & \text{if } x = 0, \\ +\infty & \text{if } x \neq 0 \end{cases} \quad \text{and} \quad \partial(\langle 0, \cdot \rangle^*)(x) = \begin{cases} H & \text{if } x = 0, \\ \emptyset & \text{if } x \neq 0. \end{cases} \tag{2.107}$$

The operator A in Theorem 2.4.13 can thus be characterized as follows:

$$\begin{aligned} A(p) &= -\begin{pmatrix} D_1 \\ \vdots \\ D_m \end{pmatrix} \partial(\langle 0, \cdot \rangle^*)(-(D_1^* \cdots D_m^*)(p)) \\ &= \begin{cases} \left\{ -\begin{pmatrix} D_1 q \\ \vdots \\ D_m q \end{pmatrix}, \ q \in H \right\} & \text{if } \sum_{i=1}^m D_i^* p_i = 0, \\ \emptyset & \text{if } \sum_{i=1}^m D_i^* p_i \neq 0. \end{cases} \end{aligned}$$

Since the conjugate function of $(v_1, \ldots, v_m) \mapsto \sum_{i=1}^m g_i(v_i)$ is given by $p \mapsto (g_1^*(p_1), \ldots, g_m^*(p_m))$ we get

$$0 \in A(p) + B(p) \Leftrightarrow \begin{cases} \sum_{i=1}^m D_i^* p_i = 0 \\ 0 \in -\begin{pmatrix} D_1 q \\ \vdots \\ D_m q \end{pmatrix} + \begin{pmatrix} \partial g_1^*(p_1) \\ \vdots \\ \partial g_m^*(p_m) \end{pmatrix} \text{ for some } q \in H. \end{cases} \tag{2.108}$$

From the definition of A and B given above it is clear that finding an element $p \in H_1 \times \cdots \times H_m$ such that $0 \in A(p) + B(p)$ is equivalent to solving the dual problem of (2.102) using the same definition of the problem (D) as presented in the general setting of Section 2.2. In particular, if we apply the construction of Section 2.2 to the operators $g = \langle 0, \cdot \rangle$, $\Phi(p) = \sum_{i=1}^m g_i(p_i)$ and the linear mapping $D = (D_1, \ldots, D_m)$, we obtain together with (2.107) that

$$\max_{p_1, \ldots, p_m} \{ -\sum_{i=1}^m g_i^*(p_i) \} \quad \text{s.t.} \quad \sum_{i=1}^m D_i^* p_i = 0 \tag{2.109}$$

is an equivalent characterization of problem (2.108).

Remark 2.4.14 *Let us briefly show the construction of the dual problem* (2.109)*,or equivalently of* (2.108)*, using the perturbation theory presented in Section 2.2. We define the perturbation functional* φ *as follows*

$$\varphi(u, p_1, \ldots, p_m) = \sum_{i=1}^m g_i(D_i u + p_i)$$

and thus the corresponding dual problem is given by

$$\max_{p_1,\dots,p_m} \{-\varphi^*(0,p_1,\dots,p_m)\}.$$

We have

$$\varphi^*(u,p_1,\dots,p_m) = \sup_{u'} \big(\langle u,u'\rangle + \sum_{i=1}^{m} \sup_{p_i'}\{\langle p_i,p_i'\rangle - g_i(D_iu'+p_i')\}\big)$$

so that

$$\varphi^*(0,p_1,\dots,p_m) = \sup_{u'} \big(\sum_{i=1}^{m} \sup_{p_i'}\{\langle p_i,p_i'\rangle - g_i(D_iu'+p_i')\}\big).$$

With a change of variables we obtain the dual problem

$$\max_{p_1,\dots,p_m} \{-\varphi^*(0,p_1,\dots,p_m)\} = \max_{p_1,\dots,p_m} \{ \inf_{u'}\{\langle \sum_{i=1}^{m} D_i^*p_i, u'\rangle\} - \sum_{i=1}^{m} g_i^*(p_i)\},$$

which is clearly equivalent to (2.109).

CHAPTER 3

Application to image denoising

3.1 Introduction

A classical problem in image processing is the *denoising* of images. In recent years, variational models were successfully applied in image denoising. The main objective of this chapter is to apply the operator splitting and Bregman techniques of Chapter 2 to these optimization problems.

This chapter is structured as follows: We start with continuous considerations in Section 3.2. Based on this, we introduce discrete models for Gaussian noise removal in Section 3.3. The two main algorithms from Chapter 2 which are useful for the Besov norm model and the Rudin-Osher-Fatemi (ROF) model, the alternating split Bregman method and the forward-backward splitting method, are discussed in Section 3.4. Furthermore, we give a geometrical intuition of both algorithms by characterizing the forward-backward splitting method as a gradient descent reprojection technique and the alternating split Bregman method as a projected version of a Newton method. It turns out that both algorithms can be written in terms of shrinkage operations. In particular, we will see that in a special setting based on Parseval frames the two algorithms are equivalent and that there exists a connection to classical wavelet shrinkage. In Section 3.5, we compare the forward-backward splitting algorithm and the alternating split Bregman algorithm numerically with multistep methods for the minimization of the ROF model and examine the choice of the step length in the alternating split Bregman algorithm. Section 3.6 shows two possible ways how the ROF model can be improved to yield better denoising results. Finally, we show in Section 3.7 the advantages of the alternating split Bregman method with multiple splittings when applied to more complicated problems, namely, the denoising and deblurring of images corrupted by Poisson noise.

3.2 Continuous denoising models

From a statistical point of view, solving a denoising problem means to find the image which is the most likely one, given the noisy image f as well as

i) the noise statistics

ii) the probability distribution on the space of images.

The noise statistics can often be described or estimated but it is in most cases not clear what the statistics of the space of images should be. In fact, it will depend heavily on the specific application. Nevertheless, it is necessary to make assumptions about the probability of an image because this *prior information* allows us to find our restored image via *Bayesian statistical inference*, i.e, the MAP (maximum a posteriori) approach, cf., e.g., [66, Section 2.4] and the references therein. To illustrate this briefly, we skip the technical details and simply write $p(u)$ for the probability assigned to an image u. The noise statistics is given by $p(f|u)$ which gives the probability that the noise process leads to the noisy image f given an original image u. So, using Bayes' rule, the probability $p(u|f)$ fulfills

$$p(u|f) = \frac{p(f|u)p(u)}{p(f)}.$$

Hence, the MAP approach yields

$$\hat{u} = \operatorname*{argmax}_u \{p(f|u)p(u)\}$$

or equivalently

$$\hat{u} = \operatorname*{argmin}_u \{-\log(p(f|u)) - \log(p(u))\}. \tag{3.1}$$

In this paper, we are only interested in two important classes of noise, namely additive white Gaussian and Poisson noise. Let us calculate the first term $-\log(p(f|u))$ for these kinds of noise.

Additive white Gaussian noise. The assumption of this noise model is that each point of the original image is corrupted by adding a realization of white noise Gaussian random variable with global standard deviation σ to the image. We assume for the sake of simplicity that our image is defined on $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$ and skip most of the analytical details. Then, we have for each point $u(x,y)$ and $f(x,y)$ of the original image and the noisy image, respectively, that

$$p(f(x,y)|u(x,y)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(u(x,y)-f(x,y))^2}{2\sigma^2}}.$$

Our assumption of independence implies that

$$p(f|u) \quad = \quad \int_\Omega \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(u(x,y)-f(x,y))^2}{2\sigma^2}} \, dxdy$$

and we obtain

$$-\log(p(f|u)) = \frac{1}{2\sigma^2}\int_\Omega (u(x,y)-f(x,y))^2\,dxdy + \log(\sigma\sqrt{2\pi}).$$

After dropping the constant and after rescaling, we see that this essentially corresponds to the squared L_2-distance between f and u, i.e.,

$$-\log(p(f|u)) \leftrightarrow \frac{1}{2}\|u-f\|^2_{L_2(\Omega)}, \tag{3.2}$$

cf., [26].

Poisson noise. The probability density function of the *Poisson distribution* is given by

$$p(k) = \begin{cases} \frac{\beta^k e^{-\beta}}{k!} & k \in \mathbb{N}\cup\{0\}, \\ 0 & \text{otherwise} \end{cases}$$

for some parameter $\beta \geq 0$. So, this is a discrete distribution with support $\mathbb{N}\cup\{0\}$. Degrading an original image $u : \Omega \to \mathbb{R}$ with Poisson noise means the following: The probability that the noisy image f has the value $k \in \mathbb{N}\cup\{0\}$ at the point $(x,y) \in \Omega$ is given by

$$p(f(x,y)=k) = \frac{\beta^k e^{-\beta}}{k!}$$

with $\beta := u(x,y)$. So, the probability that the pixel value $f(x,y) \in \mathbb{N}\cup\{0\}$ originates from $u(x,y) \in \mathbb{R}$ is given by

$$p(f(x,y)|u(x,y)) = \frac{(u(x,y))^{f(x,y)}e^{-u(x,y)}}{f(x,y)!}.$$

Note that by definition we need $u(x,y) \geq 0$ and $f(x,y) \in \mathbb{N}\cup\{0\}$. For the whole image, we thus have

$$p(f|u) = \int_\Omega \frac{(u(x,y))^{f(x,y)}e^{-u(x,y)}}{f(x,y)!}\,dxdy$$

and

$$-\log(p(f|u)) = -\int_\Omega f(x,y)\log u(x,y) - u(x,y) - \log(f(x,y)!)\,dxdy.$$

Since the corrupted image f is given, we can drop the last term and obtain

$$-\log(p(f|u)) \leftrightarrow \int_\Omega u(x,y) - f(x,y)\log u(x,y)\,dxdy, \tag{3.3}$$

cf., [5, 26]. Up to an additive constant, this term equals

$$I(f,u) := \int_\Omega f(x,y)\log\frac{f(x,y)}{u(x,y)} - f(x,y) + u(x,y)\,dxdy$$

which is known as Csizár's *I-divergence* [81] or as the *generalized Kullback-Leibler* divergence. Observe that the I-divergence is the Bregman distance of the function

$$F(u) = \int_\Omega u(x,y)\log u(x,y) - u(x,y)\,dxdy,$$

which means that

$$I(f,u) = F(f) - F(u) - \langle p, f-u\rangle, \quad \text{where} \quad p \in \partial F(u).$$

Therefore, it shares the useful properties of a Bregman distance, in particular, $I(f,u) \geq 0$.

We assume now that the second term in (3.1) has the form $-\log(p(u)) = \Phi(Du)$ for a proper, convex and l.s.c. function Φ and a bounded linear mapping D. Then, our variational denoising techniques will consist in solving a minimization problem of the form (P) defined in Section 2.2

$$\hat{u} = \operatorname*{argmin}_u \big\{ \underbrace{g(u)}_{\text{data-fitting term}} + \underbrace{\Phi(Du)}_{\text{regularization term}} \big\}, \tag{3.4}$$

where g will either be given by (3.2) or by (3.3). As illustrated above, the job of the data-fitting term g is to keep the resulting image "close" to the given noisy image f. The regularization term $\Phi \circ D$, on the other hand, represents our assumptions on typical properties of the denoised image. These two terms are combined additively and the coupling is steered by means of a so-called regularization parameter $\lambda > 0$, i.e., Φ has the form $\Phi = \lambda\tilde{\Phi}$.

From the various methods to choose the regularization term, we focus here on a *Besov norm* regularization and the *total-variation* (TV) regularization. Other approaches are outlined in Section 3.6. In the following, we restrict our attention to Gaussian noise. Before turning to the discrete setting, let us first consider the case where images are real-valued functions of some Banach space. For the sake of simplicity, we will assume that these Banach spaces are defined on $\mathbb{R}^2$ but the same holds true for domains like $[0,1]\times[0,1]$, for example.

Besov norm regularization

The *Besov norm model* we consider here, cf., [88], has the form

$$\operatorname*{argmin}_{u\in B^1_{1,1}(\mathbb{R}^2)} \{\frac{1}{2}\|u-f\|^2_{L_2(\mathbb{R}^2)} + \lambda\|u\|_{B^1_{1,1}(\mathbb{R}^2)}\}, \tag{3.5}$$

where the *Besov norm* $B^1_{1,1}(\mathbb{R}^2)$ can be defined in several ways. Let us first consider the characterization via the *smoothness modulus*. The first-order L_1 smoothness modulus of a function u on $\mathbb{R}^2$ is given by

$$w_1(u,t)_1 = \sup_{0\leq h\leq t} \|u(\cdot+h) - u\|_{L_1(\mathbb{R}^2)},$$

for some $t \geq 0$. A function u belongs to the *Besov space* $B^1_{1,1}(\mathbb{R}^2)$ if

$$\|u\|_{B^1_{1,1}(\mathbb{R}^2)} := \|u\|_{L_1(\mathbb{R}^2)} + \|u\|_{b^1_{1,1}(\mathbb{R}^2)} < +\infty, \tag{3.6}$$

where

$$\|u\|_{b^1_{1,1}(\mathbb{R}^2)} := \int_0^\infty \frac{w_1(u,t)_1}{t^2}\, dt. \tag{3.7}$$

The importance of problem (3.5) for image processing stems from the fact that the Besov norm can also be defined via the ℓ_1-norm of wavelet coefficients. To this end, we assume that for a wavelet function ψ, the set

$$\{\psi^{(j,k)} := 2^{-j/2}\psi(2^{-j}\cdot - k)\,,\; j,k \in \mathbb{Z}\}$$

forms an orthonormal basis of $L_2(\mathbb{R}^2)$, cf., Appendix A. If, in addition, ψ is sufficiently smooth and has more than one vanishing moment, then the ℓ_1-norm of the wavelet coefficients is equivalent to the Besov norm $B^1_{1,1}(\mathbb{R}^2)$ defined in (3.6), i.e.,

$$\|u\|_{B^1_{1,1}(\mathbb{R}^2)} \sim \|(\langle u, \psi^{(j,k)}\rangle)_{j,k\in\mathbb{Z}}\|_{\ell_1} := \sum_{j,k\in\mathbb{Z}} |\langle u, \psi^{(j,k)}\rangle|. \tag{3.8}$$

For more details, we refer, e.g., to [58, 73, 89, 152].

It turns out, however, that many interesting images are not in $B^1_{1,1}(\mathbb{R}^2)$, e.g., characteristic functions of simple sets do not belong to this Besov space, see Fig. 3.1.

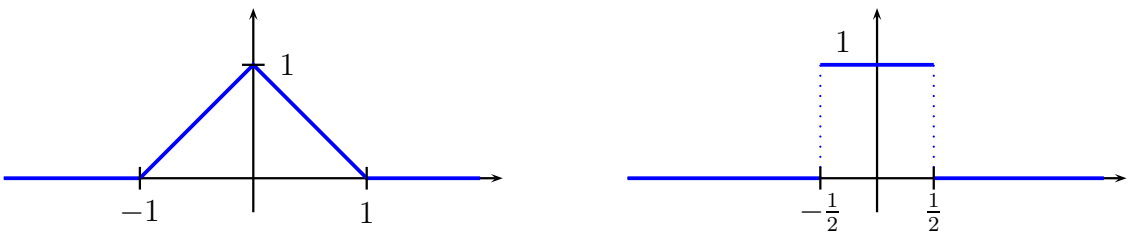


Figure 3.1: The hat function (left) lies in $B^1_{1,1}(\mathbb{R})$. The characteristic function of the set $I = [-\frac{1}{2}, \frac{1}{2}]$ (right) on the other hand does not since the integral in (3.7) behaves as $\int \frac{1}{t} dh$.

One appropriate way to enlarge the space of images is to consider the TV semi-norm. Note that, strictly speaking, this semi-norm does also not give rise to an appropriate space for many images, cf., [122].

TV regularization

Let us consider the Banach space of functions of *bounded variation*, denoted by $BV(\mathbb{R}^2)$. These are functions which fulfill

$$\|u\|_{BV(\mathbb{R}^2)} := \|u\|_{L_1(\mathbb{R}^2)} + \|u\|_{TV(\mathbb{R}^2)} < +\infty,$$

where the *total variation (TV) semi-norm* $\|\cdot\|_{TV(\mathbb{R}^2)}$ is defined as

$$\|\cdot\|_{TV(\mathbb{R}^2)} = \sup\{\int_{\mathbb{R}^2} u \operatorname{div} g \, dxdy \,:\, g \in \mathcal{C}_c^1(\mathbb{R}^2, \mathbb{R}^2) \text{ and } |g| \leq 1\}, \tag{3.9}$$

with $\mathcal{C}_c^1(\mathbb{R}^2, \mathbb{R}^2)$ being the continuously differentiable functions with compact support mapping $\mathbb{R}^2$ onto $\mathbb{R}^2$, see, e.g., [115, 157]. Note that the functions shown in Fig. 3.1 lie in this space. The total variation functional has a shorter form for functions u which are in the *Sobolev space* $W_{1,1}^1(\mathbb{R}^2)$, i.e., for which the following norm exists

$$\|u\|_{W_{1,1}^1(\mathbb{R}^2)} = \|u\|_{L_1(\mathbb{R}^2)} + \|\frac{\partial u}{\partial x}\|_{L_1(\mathbb{R}^2)} + \|\frac{\partial u}{\partial y}\|_{L_1(\mathbb{R}^2)},$$

where the partial derivatives are understood in the weak sense, cf., e.g., [2]. In this special case we have that

$$\|u\|_{TV(\mathbb{R}^2)} = \int_{\mathbb{R}^2} |\nabla u(x)| \, dxdy. \tag{3.10}$$

The following embeddings between the above Banach spaces hold true:

$$B_{1,1}^1(\mathbb{R}^2) \subset W_{1,1}^1(\mathbb{R}^2) \subset BV(\mathbb{R}^2).$$

More details and related image spaces can be found in [9, 88, 157].

Based on (3.9), we consider the *Rudin-Osher-Fatemi* (ROF) model [187, 215]

$$\operatorname*{argmin}_{u \in BV(\mathbb{R}^2)} \frac{1}{2}\|u - f\|_{L_2(\mathbb{R}^2)}^2 + \lambda \|u\|_{TV(\mathbb{R}^2)}. \tag{3.11}$$

This model is very popular and is used alongside with other techniques and with many modifications in various algorithms, cf., Section 3.6. We will also apply it for the following more advanced problems of inpainting and the denoising of tensor-valued data. The reason for the success of the ROF model is that it is an edge-preserving image restoration method, i.e., it does not "smooth" edges away as methods with L_2 regularization terms typically do.

Finally, note that a certain relation between the ROF and the Besov norm model via so-called near minimizers was studied in [18, 74].

3.3 Discrete models for Gaussian noise removal

In the following, we restrict our attention to a discrete setting. Basically, we consider discretizations of continuous gray value images on a regular grid so that our digital images are mappings $\{1, \ldots, n\} \times \{1, \ldots, n\} \to \mathbb{R}$. We often reshape such an image/matrix $F \in \mathbb{R}^{n,n}$ columnwise into a vector $f \in \mathbb{R}^N$ with length $N = n^2$. The restriction to square images is for notational convenience only and does not influence our results. If not stated

otherwise the multiplication of vectors, their square root etc. are meant componentwise. Furthermore, we denote by $\|\cdot\|_p$, $p = \{1, 2, \ldots, \infty\}$, the ℓ_p-norms in the Hilbert space $\mathbb{R}^N$. The discrete problems corresponding to (3.5) and (3.11) will have the form

$$\operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{\frac{1}{2}\|u - f\|_2^2 + \Phi(Du)\big\} \tag{3.12}$$

with appropriate choices for the matrix $D \in \mathbb{R}^{M,N}$ and the functional $\Phi : \mathbb{R}^M \to \mathbb{R}$.

Note that many of the results we present for problem (3.12) also hold true for the related deblurring problem

$$\operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{\frac{1}{2}\|Ku - f\|_2^2 + \Phi(Du)\big\} \tag{3.13}$$

with a blur matrix $K \in \mathbb{R}^{N,N}$, cf., e.g., [67, 80, 83, 101, 120].

Discrete Besov norm model

The basic idea is to make use of the characterization (3.8) of the Besov norm in terms of the ℓ_1-norm of wavelet coefficients. We have already stated in (3.8) that under appropriate conditions on an orthogonal wavelet basis the Besov norm can be written as the ℓ_1-norm of the wavelet coefficients. In the discrete setting, consider the *orthogonal* matrix $Q \in \mathbb{R}^{N,N}$ having as rows the filters of orthogonal wavelets (and scaling functions) up to a certain level. Then, problem (3.5) has the discrete counterpart

$$\begin{aligned} \hat{u} &= \operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{\frac{1}{2}\|u - f\|_2^2 + \lambda\|Qu\|_1\big\} \\ &= \operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{\frac{1}{2}\|Qu - Qf\|_2^2 + \lambda\|Qu\|_1\big\}. \end{aligned} \tag{3.14}$$

Setting $c := Qf$ and using the orthogonality of Q, we can solve

$$\hat{d} = \operatorname*{argmin}_{d\in\mathbb{R}^N} \big\{\frac{1}{2}\|d - c\|_2^2 + \lambda\|d\|_1\big\} \tag{3.15}$$

and obtain $\hat{u} = Q^{\mathrm{T}}\hat{d}$. Problem (3.15) can be solved by the known *soft shrinkage procedure* (3.22). We get $\hat{u}$ by a wavelet transform Q followed by soft shrinkage of the wavelet coefficients and the inverse wavelet transform Q^{T}. The whole procedure is called wavelet shrinkage.

However, for image processing tasks like denoising or segmentation, ordinary orthonormal wavelets are not suited due to their lack of redundancy and translational invariance which leads to visible artifacts. So, we choose for $D_1 := W \in \mathbb{R}^{M,N}$, $M = pN$, where W is a matrix whose rows form a *Parseval frame* of $\mathbb{R}^N$. The matrix W being a Parseval frame means that

$$W^{\mathrm{T}}W = I \quad \text{but in general} \quad WW^{\mathrm{T}} \neq I.$$

Here, p is three times the wavelet decomposition level plus one for the rows belonging to the scaling function filters on the coarsest scale. For a more detailed description, see Appendix A.

Motivated by (3.8) we take the ℓ_1-norm of the coefficient vector Wu, i.e., the functional $\Phi_1 : \mathbb{R}^M \to \mathbb{R}$ is defined as follows

$$\Phi_1(d) := \|\Lambda d\|_1 \quad \text{with} \quad \Lambda := \operatorname{diag}(\lambda_j)_{j=1}^M, \quad \lambda_j \geq 0.$$

Discrete ROF model

Let us now discretize the TV functional. There are many methods to discretize the gradient operator. We will consider here a discretization based on simple forward differences. Starting with the forward difference matrix

$$D_f := \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix} \in \mathbb{R}^{n,n}, \tag{3.16}$$

we approximate the partial derivative operators by

$$\begin{aligned} D_x &:= I \otimes D_f, \\ D_y &:= D_f \otimes I, \end{aligned} \tag{3.17}$$

where $K_1 \otimes K_2$ denotes the *Kronecker product* of the matrices K_1 and K_2. Hence, we define the discrete gradient operator as

$$D_2 := \begin{pmatrix} D_x \\ D_y \end{pmatrix}. \tag{3.18}$$

This discretization was, e.g., used in Chambolle's paper [56]. According to (3.10), we have to compute the absolute value of the gradient, i.e., in our discrete setting $((D_x u)^2 + (D_y u)^2)^{\frac{1}{2}}$, and to sum up over all points. Consequently, we define Φ_2 as follows:

$$\Phi_2(d) := \|\tilde{\Lambda}\,|d|\,\|_1 \tag{3.19}$$

with $\tilde{\Lambda} := \operatorname{diag}(\tilde{\lambda}_j)_{j=1}^N$ and $|d| := \Big(\|\mathbf{d}_j\|_2\Big)_{j=1}^N$ for the tuples $\mathbf{d}_j := (d_j, d_{j+N})$.

Observe that other choices to discretize the gradient operator are possible: Substituting the forward differences in (3.16) with central differences introduces checkerboard patterns. However, in the context of discretizing derivatives for diffusion problems solved via partial differential equations, the problem of checkerboard artifacts was dealt with successfully in [190, 196, 207] by adding an additional filter.

In [161, 224], the following *consistent* finite difference discretization of $|\nabla u|$ was proposed:

$$|\nabla u| \approx \left(\left((H_0 \otimes H_1)\, u \right)^2 + \left((H_1 \otimes H_0)\, u \right)^2 + \left((H_1 \otimes H_1)\, u \right)^2 \right)^{\frac{1}{2}}, \tag{3.20}$$

where H_0 and H_1 are again the matrices of the Haar filters $\frac{1}{2}[1\ \ 1]$ and $\frac{1}{2}[1\ \ -1]$, respectively. We have given an intuitive numerical example for the performance of the ROF model containing forward difference, central differences and the discretization (3.20) in [194]. We mention further [205, 224, 226] for relations between the ROF and the Besov norm model in the discrete setting.

Finally, we remark that there exist anisotropic ROF models which use $|u_x|+|u_y|$ instead of $|\nabla u|$, i.e., $\|D_2 u\|_1$ in the discrete setting. Based on [22], its modifications were used in [100, 196, 203, 207] for images with a special directional structure, e.g., sharp corners.

Proximation with respect to Φ_i and Φ_i^*

When applying splitting algorithms to (3.12), we have to solve the proximation problem with respect to Φ_i and Φ_i^*, $i = 1, 2$. This can be done in a straightforward way using fast shrinkage operations. Indeed, one reason why splitting methods have become so popular recently in image processing is that prox_{Φ_i} and $\mathrm{prox}_{\Phi_i^*}$, $i = 1, 2$, can be computed in a very efficient way.

A short calculation shows that for any $c \in \mathbb{R}^M$ we have

$$\mathrm{prox}_{\Phi_1}(c) = \mathcal{T}_\Lambda(c), \quad \mathrm{prox}_{\Phi_2}(c) = \widetilde{\mathcal{T}}_{\tilde{\Lambda}}(c). \tag{3.21}$$

Here, $\mathcal{T}_\Lambda$ denotes the *soft shrinkage* or *soft thresholding* function given componentwise by

$$\mathcal{T}_{\lambda_j}(c_j) := \begin{cases} 0 & \text{if } |c_j| \le \lambda_j, \\ c_j - \lambda_j \operatorname{sgn}(c_j) & \text{if } |c_j| > \lambda_j, \end{cases} \tag{3.22}$$

or equivalently

$$\mathcal{T}_{\lambda_j}(c_j) = \frac{1}{2}\left((c_j - \lambda_j) + |c_j - \lambda_j| + (c_j + \lambda_j) - |c_j + \lambda_j| \right), \qquad j = 1, \ldots, M.$$

The mapping $\widetilde{\mathcal{T}}_{\tilde{\Lambda}}$ denotes the *coupled shrinkage* or *coupled thresholding* function, compare [57, 161, 224, 226], which is defined for each part $\mathbf{c}_j = (c_{j+kN})_{k=0}^{p-1}$ of a vector $c \in \mathbb{R}^{pN}$ as

$$\widetilde{\mathcal{T}}_{\tilde{\lambda}_j}(\mathbf{c}_j) := \begin{cases} 0 & \text{if } \|\mathbf{c}_j\|_2 \le \tilde{\lambda}_j, \\ \mathbf{c}_j - \tilde{\lambda}_j \mathbf{c}_j / \|\mathbf{c}_j\|_2 & \text{if } \|\mathbf{c}_j\|_2 > \tilde{\lambda}_j. \end{cases} \tag{3.23}$$

The dual problem of (3.12) has the form

$$\hat{b} = \operatorname*{argmin}_{b \in \mathbb{R}^M} \{ \frac{1}{2} \|D^{\mathrm{T}} b - f\|_2^2 + \Phi^*(b) \} \tag{3.24}$$

and the primal and dual solutions $\hat{u}$ and $\hat{b}$, respectively, are connected via

$$\hat{u} = f - D^{\mathrm{T}}\hat{b},$$

see also, [56, 200]. To calculate the conjugate functions of Φ_i^*, $i = 1, 2$, we need the notion of an *indicator function*. For a set $C \in \mathbb{R}^M$ the indicator function ι_C is given by

$$\iota_C(d) := \begin{cases} 0 & \text{if } d \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

The functions Φ_i have the nice property of being *positively homogeneous* (or one-homogeneous), i.e., for all $\alpha \geq 0$ and $d \in \mathbb{R}^M$ we have $\Phi_i(\alpha d) = \alpha \Phi_i(d)$, $i = 1, 2$. It is a classical result in convex analysis that finite positively homogeneous convex functions and the indicator function of nonempty, closed, convex and bounded sets are conjugate to each other, see [181, Theorem 13.2]. More precisely, we have that

$$\Phi_i^*(d) = \iota_{C_i}, \tag{3.25}$$

with $C_i = \{d \in \mathbb{R}^M : \langle d, x\rangle \leq \Phi_i(x)\,, \forall x \in \mathbb{R}^M\}$ for $i = 1, 2$. Using that the dual norm of the ℓ_1-norm is the ℓ_∞-norm and the ℓ_2-norm is self-dual, one can show, see, e.g., [90] that for our setting

i) $C_1 = \{d \in \mathbb{R}^M : |d_j| \leq \lambda_j, \;\; j = 1, \ldots, M\}$,

ii) $C_2 = \{d \in \mathbb{R}^M : \|\mathbf{d}_j\|_2 \leq \tilde{\lambda}_j, \;\; j = 1, \ldots, N\}$.

In conclusion, the dual problem (3.24) has the form

$$\hat{b} = \underset{b \in \mathbb{R}^M}{\operatorname{argmin}} \{\frac{1}{2}\|D_i^{\mathrm{T}} b - f\|_2^2\} \quad \text{s.t.} \quad b \in C_i, \quad i = 1, 2, \tag{3.26}$$

see also [56, 90, 200]. Observe that the dual problem is a linear least squares problem with relatively simple constraints. In the case i), we have, e.g., symmetric box constraints.

The proximity operator of Φ_i^* has the form

$$\operatorname{prox}_{\Phi_i^*}(c) = \operatorname{prox}_{\iota_{C_i}}(c) = P_{C_i}(c),$$

where $P_{C_i}(c)$ denotes the projection of a vector $c \in \mathbb{R}^M$ onto the set C_i for $i = 1, 2$. A short calculation shows that this projection can also be expressed in terms of the soft and coupled shrinkage operators defined in (3.22) and (3.23), respectively:

$$\operatorname{prox}_{\Phi^*}(c) = c - \mathcal{T}(c) \tag{3.27}$$

for $(\Phi, \mathcal{T}) \in \{(\Phi_1, \mathcal{T}_\Lambda), (\Phi_2, \widetilde{\mathcal{T}}_{\tilde{\Lambda}})\}$ and all $c \in \mathbb{R}^M$. As we will see in the next sections, it will be crucial for the performance of most of the algorithms we deal with in this paper that the projections onto C_1 and C_2 can be computed in this fast way.

Note that, in general, we will work with a single regularization parameter, i.e., $\Lambda = \lambda I$ or $\tilde{\Lambda} = \lambda I$ for some $\lambda \geq 0$.

3.4 DRS and FBS for Gaussian noise removal

We will now apply the algorithms defined in Chapter 2 to the discrete denoising problem (3.12). In particular, we consider the Douglas-Rachford splitting/alternating split Bregman algorithm applied to the primal problem (3.12) as well as the forward-backward splitting applied to the corresponding dual problem (3.24).

3.4.1 DRS algorithm

Consider the DRS/alternating split Bregman algorithm (2.76)-(2.78) with $g(u) := \frac{1}{2}\|u - f\|_2^2$ and the Besov norm or TV regularizer. It has the form

$$u^{(k+1)} = \operatorname*{argmin}_{u \in H_1} \big\{ \frac{1}{2}\|u - f\|_2^2 + \frac{1}{2\gamma}\|b^{(k)} + Du - d^{(k)}\|_2^2 \big\}, \tag{3.28}$$

$$d^{(k+1)} = \operatorname*{argmin}_{d \in H_2} \big\{ \Phi(d) + \frac{1}{2\gamma}\|b^{(k)} + Du^{(k+1)} - d\|_2^2 \big\}, \tag{3.29}$$

$$b^{(k+1)} = b^{(k)} + Du^{(k+1)} - d^{(k+1)}, \tag{3.30}$$

for $(\Phi, D) \in \{(\Phi_1, D_1), (\Phi_2, D_2)\}$. Theorem 2.4.7 and Corollary 2.4.10 imply the convergence of $(b^{(k)})_{k\in\mathbb{N}}$ and $(u^{(k)})_{k\in\mathbb{N}}$, respectively. To minimize the quadratic functional in (3.28), we have to solve the following linear system of equations

$$u^{(k+1)} = (\gamma I + D^{\mathrm{T}}D)^{-1}\big(\gamma f + D^{\mathrm{T}}(d^{(k)} - b^{(k)})\big).$$

In the Besov norm case, this is easy to solve since $D_1^{\mathrm{T}} D_1 = I$ but for the matrix $D = D_2$ this is the hardest subproblem of the algorithm and various methods can be used to compute a solution or an approximate solution. In [120], e.g., Osher and Goldstein use a Gauß-Seidel step to approximate $u^{(k+1)}$ since the coefficient matrix resulting from the forward difference matrix is strictly diagonally dominant. Another possibility is to diagonalize $D_2^{\mathrm{T}} D_2$ via the matrix of the cosine-II transform. This allows us to use fast cosine algorithms and the complexity becomes $\mathcal{O}(n^2 \log n)$, cf., Appendix B. In step (3.29) we have to compute $\operatorname{prox}_{\gamma\Phi}(b^{(k)} + Du^{(k+1)})$. Here, we can use (3.21).

The following algorithm shows the alternating split Bregman algorithm (2.76)-(2.78) for the case $(\mathcal{T}, D) \in \{(\mathcal{T}_\Lambda, D_1), (\widetilde{\mathcal{T}}_\Lambda, D_2)\}$. Observe that we have changed the order in which we compute $u^{(k+1)}$. This is allowed because there are no restrictions on the choice of the starting values. The reason for this different definition is that it allows us to better compare this method to the forward-backward splitting algorithm in the following subsection.

Algorithm (DRS/Alternating split Bregman shrinkage for (3.12)**)**
Initialization: $u^{(0)} := f$, $b^{(0)} := 0$

For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} d^{(k+1)} &:= \mathcal{T}(b^{(k)} + Du^{(k)}), \\ b^{(k+1)} &:= b^{(k)} + Du^{(k)} - d^{(k+1)}, \\ u^{(k+1)} &:= (\gamma I + D^{\mathrm{T}}D)^{-1}\big(\gamma f + D^{\mathrm{T}}(d^{(k+1)} - b^{(k+1)})\big). \end{aligned} \tag{3.31}$$

Clearly, we can also define an alternating split Bregman algorithm according to the multiple splitting algorithm (2.98)-(2.101). If we again compute $u^{(k+1)}$ in the end of each iteration, we get the following algorithm:

$$\begin{aligned} v^{(k+1)} &:= \frac{1}{1+\gamma}(\gamma f + b_1^{(k)} + u^{(k+1)}), \\ d^{(k+1)} &:= \mathcal{T}(b_2^{(k)} + Du^{(k+1)}), \\ b^{(k+1)} &:= b^{(k)} + \begin{pmatrix} I \\ D \end{pmatrix} u^{(k+1)} - \begin{pmatrix} v^{(k+1)} \\ d^{(k+1)} \end{pmatrix}, \\ u^{(k+1)} &:= (I + D^{\mathrm{T}}D)^{-1}\big(v^{(k+1)} - b_1^{(k+1)} + D^{\mathrm{T}}(d^{(k+1)} - b_2^{(k+1)})\big). \end{aligned} \tag{3.32}$$

Note that in contrast to (3.31) the parameter γ does not appear in the last step of (3.32) which yields for *small* γ a better condition number of the corresponding matrix. Convergence of the sequences $(v^{(k)})_{k\in\mathbb{N}}$, $(d^{(k)})_{k\in\mathbb{N}}$ and $(b^{(k)})_{k\in\mathbb{N}}$ generated by (3.32) is guaranteed by Theorem 2.4.13. To prove the convergence of $(u^{(k)})_{k\in\mathbb{N}}$ we can either apply the same argument as in Corollary 2.4.10 and the fact that the linear operator $I + D^{\mathrm{T}}D$ is invertible or we can use directly the coupling of the variables, i.e., $u^{(k)} - v^{(k)} \to 0$ in (3.32). Furthermore, Theorem 2.4.13 and an argument very similar to the one used in Proposition 2.4.8 implies that $\hat{u}$ and $\frac{1}{\gamma}\hat{b}_2$ converge to solutions of the primal problem (3.12) and the dual problem (3.26), respectively. Our numerical experiments show that for the ROF model or the Besov norm model and for a fixed number of iterations, the above two alternating split Bregman shrinkage algorithms yield very similar results.

Remark 3.4.1 *Recall that in Section 2.4.3 we have shown the equivalence of the alternating split Bregman algorithm and the Douglas-Rachford splitting algorithm for the dual problem. It should be noted here that the Douglas-Rachford splitting method itself was first applied to image processing problems in* [78]. *An application to multi-class image labeling can be found in* [141].

3.4.2 FBS algorithm

We will now see that we obtain an interesting algorithm by applying the forward-backward splitting method (2.30) to the dual problem (3.24). Let us choose $A := \partial\Phi^*$ and $B := D(D^{\mathrm{T}} \cdot - f)$. Then the corresponding forward-backward splitting algorithm is given by

$$b^{(k+1)} = \mathrm{prox}_{\gamma\Phi^*}\left(b^{(k)} + \gamma D(f - D^{\mathrm{T}}b^{(k)})\right). \tag{3.33}$$

This algorithm converges for $0 < \gamma < 2/\|D^{\mathrm{T}}D\|_2$ which can be seen as follows: Clearly, $\partial\Phi^*$ is maximal monotone. Moreover, B is single-valued and we have

$$\|D(D^{\mathrm{T}}b_1 - f) - D(D^{\mathrm{T}}b_2 - f)\|_2 \leq \|D^{\mathrm{T}}D\|_2 \, \|b_1 - b_2\|_2,$$

i.e., B is Lipschitz continuous with constant $\|D^{\mathrm{T}}D\|_2$. So, we get by Theorem 2.3.19 that $\frac{1}{\|D^{\mathrm{T}}D\|_2}B$ is firmly nonexpansive and the convergence follows from Theorem 2.3.17.

Using the relation (3.27), we can rewrite (3.33) for $(\mathcal{T}, D) \in \{(\mathcal{T}_1, D_1), (\mathcal{T}_2, D_2)\}$ as

$$b^{(k+1)} = b^{(k)} + \gamma D(f - D^{\mathrm{T}}b^{(k)}) - \mathcal{T}\big(b^{(k)} + \gamma D(f - D^{\mathrm{T}}b^{(k)})\big). \tag{3.34}$$

To see the relation to the alternating split Bregman shrinkage algorithm defined in (3.31), we introduce the primal variable $u^{(k)} = f - D^{\mathrm{T}}b^{(k)}$ and rewrite (3.34) in the following way:

Algorithm (Forward-backward splitting shrinkage for (3.26)**)**
Initialization: $u^{(0)} := f$, $b^{(0)} := 0$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} d^{(k+1)} &:= \mathcal{T}\big(b^{(k)} + \gamma D u^{(k)}\big), \\ b^{(k+1)} &:= b^{(k)} + \gamma D u^{(k)} - d^{(k+1)}, \\ u^{(k+1)} &:= f - D^{\mathrm{T}}b^{(k+1)}. \end{aligned} \tag{3.35}$$

This algorithm can also be deduced as a simple *gradient descent reprojection algorithm* as it was done, e.g., by Chambolle [57]. Algorithm (3.35) is *not* Chambolle's semi-implicit gradient descent algorithm of [56] which we study in Appendix C.

A relation of the forward-backward splitting shrinkage method to the *Bermúdez-Moreno algorithm* of [23], which also turns out to be the above forward-backward splitting algorithm, was shown in [8].

Remark 3.4.2 *Recently, there is a lot of interest in improvements of the gradient descent reprojection algorithm by using adaptive step size strategies. This is mainly motivated by the work of Barzilai and Borwein in* [12]*. For details and numerical comparisons we refer to* [146, 234, 236].

3.4.3 Relation between DRS and FBS algorithm

We illustrate in this section that for both Besov norm and ROF denoising a special choice of the matrices D_1, D_2 and of the parameter γ gives rise to a close relation between the corresponding DRS/alternating split Bregman and the forward-backward splitting algorithms. Furthermore, we show the connection to wavelet shrinkage.

Besov norm regularization

The matrix D_1 is again a Parseval frame matrix. Equality (3.14) is still true, but the problem is no longer equivalent to (3.15). Instead of (3.15), we have to solve the constrained problem

$$\hat{d} = \operatorname*{argmin}_{d \in \mathbb{R}^M} \big\{ \frac{1}{2}\|c - d\|_2^2 + \|\Lambda d\|_1 \big\} \qquad \text{subject to } \; d \in \mathcal{R}(D_1), \tag{3.36}$$

where $\mathcal{R}(D_1)$ denotes the range of D_1. The solution in the image space is then again given by $\hat{u} = D_1\hat{d}$. The constraint in (3.36) is equivalent to $(I - D_1 D_1^{\mathrm{T}})d = 0$, i.e., d is equal to its orthogonal projection on $\mathcal{R}(D_1)$. One could penalize this condition. In the context of inpainting, this was suggested in [51].

We will show that the forward-backward splitting shrinkage algorithm and the alternating split Bregman shrinkage algorithm with $D = D_1$, $\gamma = 1$ and $\Phi = \Phi_1$ are the same. Moreover, we will see that they coincide with the following algorithm which underlines the relation to the wavelet shrinkage algorithm for orthonormal transforms.

Algorithm (Iterated frame shrinkage)
Initialization: $u^{(0)} := f$, $b^{(0)} := 0$.

For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} d^{(k+1)} &:= \mathcal{T}_\Lambda(b^{(k)} + D_1 u^{(k)}), \\ b^{(k+1)} &:= b^{(k)} + D_1 u^{(k)} - d^{(k+1)}, \\ u^{(k+1)} &:= D_1^{\mathrm{T}} d^{(k+1)}. \end{aligned} \tag{3.37}$$

The first step of the algorithm, i.e., $u^{(1)} = D_1^{\mathrm{T}}\mathcal{T}_\Lambda(D_1 f)$ is an ordinary frame shrinkage step which also appears if we skip the constraint in (3.36). In the following iterations, the algorithm differs from the usual iterated frame shrinkage in the summand $b^{(k)}$ we have to add before shrinking in $d^{(k+1)}$.

Note that in order to use the forward-backward splitting algorithm for problem (3.14), γ has to fulfill $0 < \gamma < 2/\|D_1^{\mathrm{T}} D_1\|_2$. Now $D_1^{\mathrm{T}} D_1 = I_N$, thus we have to choose γ in $(0, 2)$ and $\gamma = 1$ is an admissible choice.

Proposition 3.4.3 *For $D := D_1$ and $\gamma := 1$ the forward-backward splitting shrinkage algorithm* (3.35) *and the alternating split Bregman shrinkage algorithm* (3.31) *coincide with the iterated frame shrinkage algorithm.*

Proof: We start with $u^{(0)} := f$ and $b^{(0)} := 0$ in all three algorithms and in the way we have written them, they only differ in the third step, where we have

$$\begin{aligned} u^{(k+1)} &= \tfrac{1}{2}\big(f + D_1^{\mathrm{T}}(d^{(k+1)} - b^{(k+1)})\big) \quad \text{in algorithm (3.31)}, \\ u^{(k+1)} &= f - D_1^{\mathrm{T}} b^{(k+1)} \quad \text{in algorithm (3.35)}, \\ u^{(k+1)} &= D_1^{\mathrm{T}} d^{(k+1)} \quad \text{in the iterated frame shrinkage algorithm (3.37)}. \end{aligned}$$

We now use induction on k. Assume that $u^{(k)} = f - D_1^{\mathrm{T}} b^{(k)}$. Then, we obtain by the definition of $b^{(k+1)}$ that

$$u^{(k+1)} = f - D_1^{\mathrm{T}} b^{(k+1)} = f - D_1^{\mathrm{T}} b^{(k)} - u^{(k)} + D_1^{\mathrm{T}} d^{(k+1)} = D_1^{\mathrm{T}} d^{(k+1)},$$

so that the elements $u^{(k+1)}$ are the same for the forward-backward splitting shrinkage and the iterated frame shrinkage. Further, we see

$$\begin{aligned} \frac{1}{2}\big(f + D_1^{\mathrm{T}}(d^{(k+1)} - b^{(k+1)})\big) &= \frac{1}{2}\big(f + D_1^{\mathrm{T}} d^{(k+1)} - D_1^{\mathrm{T}} b^{(k)} - u^{(k)} + D_1^{\mathrm{T}} d^{(k+1)}\big) \\ &= D_1^{\mathrm{T}} d^{(k+1)} + \frac{1}{2}\big(f - D_1^{\mathrm{T}} b^{(k)} - u^{(k)}\big) \;=\; D_1^{\mathrm{T}} d^{(k+1)}, \end{aligned}$$

which means that the alternating split Bregman shrinkage algorithm coincides with the iterated frame shrinkage algorithm, too. □

Let us restate our result for $D := D_1$ and $\gamma := 1$ in the following diagram:

Alternating Split Bregman Shrinkage	=	Forward-Backward Splitting Shrinkage	=	Iterated Frame Shrinkage

ROF regularization

To show the relation between the alternating split Bregman algorithm and forward-backward splitting for the ROF denoising model, we replace the forward difference discretization D_2 by the Haar-filter based discretization of the absolute value of the gradient given by (3.20). We will denote the corresponding matrix by $\mathcal{H}_1$, i.e.,

$$\mathcal{H}_1 = \begin{pmatrix} H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix}.$$

Observe that $\mathcal{H}_1$ itself is not a Parseval frame but it is part of a Parseval frame: Define the corresponding "low-pass" part as $\mathcal{H}_0 = H_0 \otimes H_0$, then, $W := \begin{pmatrix} \mathcal{H}_0 \\ \mathcal{H}_1 \end{pmatrix}$ is exactly the Parseval frame in (A.13).

Using this gradient discretization, the discrete version of the ROF functional (3.11) reads

$$\operatorname*{argmin}_{u \in \mathbb{R}^N} \big\{ \frac{1}{2}\|u - f\|_2^2 + \|\tilde{\Lambda}\,|\mathcal{H}_1 u|\,\|_1 \big\}, \qquad \tilde{\Lambda} := \lambda I_N. \tag{3.38}$$

Clearly, we can apply the forward-backward splitting shrinkage algorithm (3.35) to this problem. It is readily seen that, using Parseval frames, we obtain equivalent versions of this forward-backward splitting shrinkage algorithm by means of the alternating split Bregman shrinkage algorithm (3.31) and the iterated frame shrinkage algorithm (3.37). To

this end, let us define the proper, convex and l.s.c. functional $\tilde{\Phi}_2$ which differs from Φ_2 in that the first part of the input vector is neglected, i.e.,

$$\tilde{\Phi}_2(c) = \|\tilde{\Lambda}\,|c_1|\,\|_1, \qquad \text{for } c = (c_0, c_1) \in \mathbb{R}^N \times \mathbb{R}^{3N}.$$

Now, we can rewrite (3.38) in the form

$$\operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{ \frac{1}{2}\|u - f\|_2^2 + \tilde{\Phi}_2(Wu) \big\}. \tag{3.39}$$

Note that in order to use forward-backward splitting, γ has to fulfill $0 < \gamma < 2/\|W^{\mathrm{T}}W\|_2$. Now $W^{\mathrm{T}}W = I_N$, thus we have to choose γ in $(0, 2)$ and $\gamma = 1$ is an admissible choice. Setting $\gamma = 1$ means that algorithms (3.31) and (3.37) applied to (3.39) are equivalent to (3.35) applied to (3.38), cf. Section 3.4.3. It is clear that for any $c = (c_0, c_1) \in \mathbb{R}^N \times \mathbb{R}^{3N}$, we have $\operatorname{prox}_{\tilde{\Phi}_2^*}(c) = (c_0, c_1 - \widetilde{\mathcal{T}}_{\tilde{\Lambda}}(c_1))$. Hence, using the notation of the iterated frame shrinkage algorithm (3.37), we obtain the following algorithm:

Initialization: $u^{(0)} := f$, $b^{(0)} := 0$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} d_0^{(k+1)} &:= (Wu^{(k)})_0, \\ d_1^{(k+1)} &:= \widetilde{\mathcal{T}}_{\tilde{\Lambda}}\big(b^{(k)} + (Wu^{(k)})_1\big), \\ b^{(k+1)} &:= b^{(k)} + (Wu^{(k)})_1 - d_1^{(k+1)}, \\ u^{(k+1)} &:= W^{\mathrm{T}} \begin{pmatrix} d_0^{(k+1)} \\ d_1^{(k+1)} \end{pmatrix}, \end{aligned} \tag{3.40}$$

where $(Wu)_0 := \mathcal{H}_0 u$ and $(Wu)_1 := \mathcal{H}_1 u$. Note that starting with $b_0^{(0)} := 0$ all iterates $b_0^{(k)}$ remain zero vectors.

Another possibility is to apply the alternating split Bregman shrinkage algorithm (3.31) with $D := \mathcal{H}_1$ which is not equivalent to the method described above. We now give a numerical example for these two algorithms. The computations were performed in MATLAB. In Fig. 3.2 we see the result of applying the two algorithms to a noisy image. Note that we only show the resulting image for algorithm (3.40) here, since the difference to the alternating split Bregman method with $D = \mathcal{H}_1$ is marginal. We also found that the two algorithms need nearly the same number of iterations. For the above numerical experiment, we used periodic boundary conditions in the Parseval frame, concerning Neumann boundary conditions, see, e.g., [61].

3.4.4 Geometrical interpretation of DRS and FBS algorithm

The forward-backward splitting shrinkage algorithm (3.35) can easily be understood geometrically. It is simply a gradient descent reprojection method with a fixed step length. In

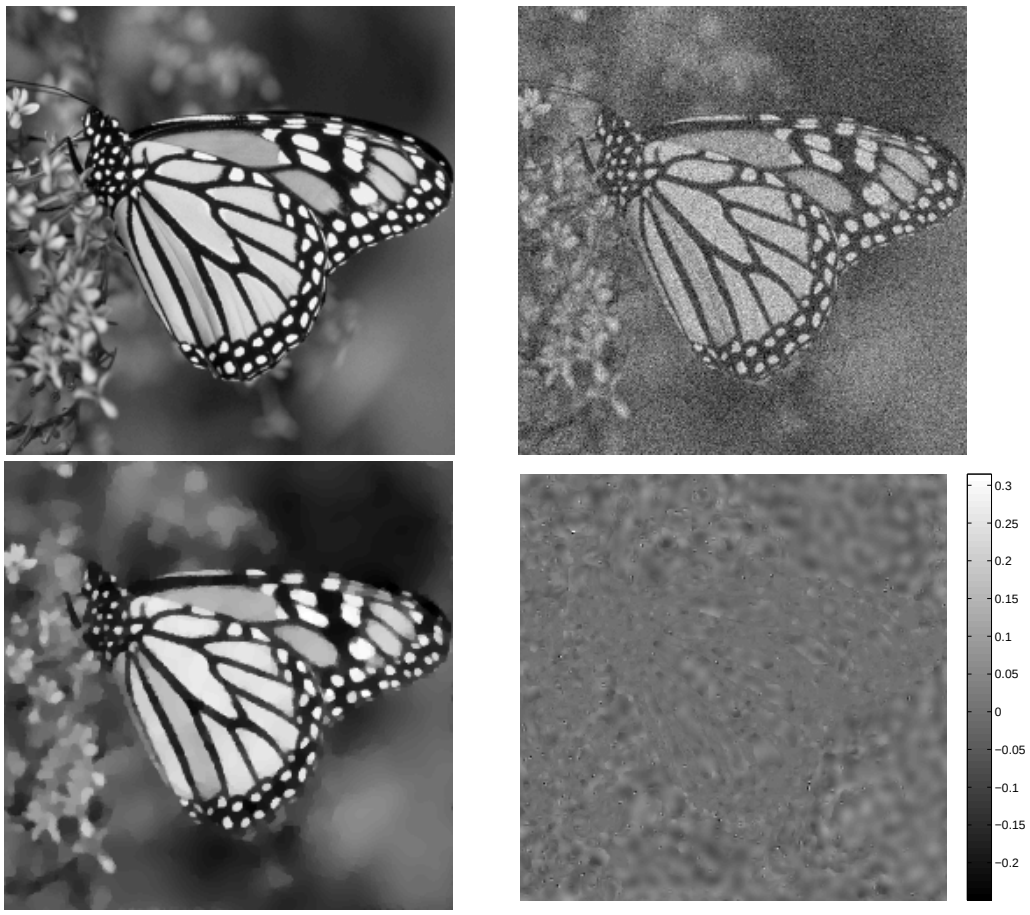


Figure 3.2: Comparison of algorithm (3.40) and algorithm (3.31) with $D = \mathcal{H}_1$. Stopping criterion: $\|u^{(k+1)} - u^{(k)}\|_\infty < 0.5$. *Top left*: Original image. *Top right*: Noisy image (white Gaussian noise with standard deviation 25). *Bottom left*: Algorithm (3.40), $\lambda = 70$, (53 iterations). *Bottom right*: Difference to algorithm (3.31) with $D = \mathcal{H}_1$, (53 iterations).

each iteration we first perform a gradient descent step with step length γ. Then, we reproject onto the feasible set C_1 or C_2. Interpreting the alternating split Bregman shrinkage (3.31) is not that straightforward. Let us consider as in (2.91) - (2.93) the slightly different version where we change the order of the first and the third step in (3.31). We obtain the algorithm

$$\begin{aligned} d^{(k+1)} &:= \mathcal{T}(b^{(k)} + Du^{(k)}), \\ u^{(k+1)} &:= (\gamma I + D^{\mathrm{T}}D)^{-1}\big(\gamma f + D^{\mathrm{T}}(d^{(k+1)} - b^{(k)})\big), \\ b^{(k+1)} &:= b^{(k)} + Du^{(k)} - d^{(k+1)}, \end{aligned} \tag{3.41}$$

where $(\mathcal{T}, D) \in \{(\mathcal{T}_1, D_1), (\mathcal{T}_2, D_2)\}$. Recall that in Corollary 2.4.12 it was shown that (2.91) - (2.93) is a Douglas-Rachford splitting algorithm with respect to the operators $A = \partial\Phi^*$ and $B = \partial(g^* \circ (-D^{\mathrm{T}}))$ for $\Phi \in \{\Phi_1, \Phi_2\}$. Hence, we can conclude that arguments

similar to the ones in Theorem 2.4.7 and in Theorem 2.4.9 guarantee the convergence of all the sequences generated by (3.41). As before, $\hat{u}$ and $\frac{1}{\gamma}\hat{b}$ are solutions of the primal problem (3.12) and the dual problem (3.26), respectively. Since the operator

$$B = \partial(g^* \circ (-D^{\mathrm{T}})) = \nabla(\tfrac{1}{2}\|D^{\mathrm{T}} \cdot -f\|_2^2) = D(D^{\mathrm{T}} \cdot -f)$$

is single-valued, we can equivalently consider the loose Douglas-Rachford splitting algorithm (2.33). The resolvent

$$J_{\frac{1}{\gamma}A} = (I + \tfrac{1}{\gamma}D(D^{\mathrm{T}} \cdot -f))^{-1}$$

allows us to write $J_{\frac{1}{\gamma}A}x = y$ equivalently as $(I + \frac{1}{\gamma}DD^{\mathrm{T}})^{-1}(x + \frac{1}{\gamma}Df) = y$. Thus, algorithm (2.33) has the following form:

$$\begin{aligned} p^{(k+1)} &= (I + \tfrac{1}{\gamma}DD^{\mathrm{T}})^{-1}(\mathrm{prox}_{\frac{1}{\gamma}\Phi^*}(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) + \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f) + \tfrac{1}{\gamma}Df) \\ &= (I + \tfrac{1}{\gamma}DD^{\mathrm{T}})^{-1}(\mathrm{prox}_{\frac{1}{\gamma}\Phi^*}(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) + \tfrac{1}{\gamma}DD^{\mathrm{T}}p^{(k)}). \end{aligned}$$

This yields

$$\begin{aligned} & p^{(k+1)} + \tfrac{1}{\gamma}DD^{\mathrm{T}}p^{(k+1)} = \mathrm{prox}_{\frac{1}{\gamma}\Phi^*}(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) + \tfrac{1}{\gamma}DD^{\mathrm{T}}p^{(k)} \\ \Leftrightarrow\ & p^{(k+1)} - p^{(k)} + \tfrac{1}{\gamma}DD^{\mathrm{T}}(p^{(k+1)} - p^{(k)}) = \mathrm{prox}_{\frac{1}{\gamma}\Phi^*}(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) - p^{(k)} \\ \Leftrightarrow\ & p^{(k+1)} - p^{(k)} = (I + \tfrac{1}{\gamma}DD^{\mathrm{T}})^{-1}(\mathrm{prox}_{\frac{1}{\gamma}\Phi^*}(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) - p^{(k)}), \end{aligned}$$

and thus, we see that $p^{(k+1)}$ is computed in the following way:

$$\begin{aligned} p^{(k+1)} &= p^{(k)} + (I + \tfrac{1}{\gamma}DD^{\mathrm{T}})^{-1}(\mathrm{prox}_{\frac{1}{\gamma}\Phi^*}(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) - p^{(k)}) \\ &= p^{(k)} + (I + \tfrac{1}{\gamma}DD^{\mathrm{T}})^{-1}(P_C(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) - p^{(k)}), \end{aligned} \tag{3.42}$$

where $C \in \{C_1, C_2\}$. Note that by Corollary 2.4.12 the sequence $(b^{(k)})_{k\in\mathbb{N}}$ in (3.41) is connected to $(p^{(k)})_{k\in\mathbb{N}}$ via $p^{(k)} = \frac{1}{\gamma}b^{(k)}$. Algorithm (3.42) bears a strong resemblance to a constrained version of the Newton method with Levenberg-Marquardt regularization, which is often called the *Levenberg-Marquardt method*, see [142, 154]. Let us disregard the constraint in the dual problem (3.26) for a moment, i.e, we consider the simple quadratic problem

$$\hat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^M}\{\frac{1}{2}\|D^{\mathrm{T}}x - f\|_2^2\}. \tag{3.43}$$

The Hessian matrix DD^{T} of (3.43) is not invertible and hence, we cannot apply Newton's method directly. The idea of the Levenberg-Marquardt method is to regularize the Hessian

with a unit matrix and an appropriate sequence of positive parameters $(\theta_k)_{k\in\mathbb{N}}$ to make it invertible:

$$x^{(k+1)} = x^{(k)} + (\theta_k I + DD^{\mathrm{T}})^{-1}(-D(D^{\mathrm{T}}x^{(k)} - f)). \tag{3.44}$$

Clearly, we can rewrite (3.44) as

$$x^{(k+1)} = x^{(k)} + (I + \tfrac{1}{\theta_k}DD^{\mathrm{T}})^{-1}(-\tfrac{1}{\theta_k}D(D^{\mathrm{T}}x^{(k)} - f)). \tag{3.45}$$

If we disregard the projection in (3.42), we see that the iteration has exactly the same form as the Levenberg-Marquardt algorithm (3.45). Our algorithm (3.42) incorporates the constraints of our dual minimization problem (3.26) by applying $(I + \frac{1}{\gamma}DD^{\mathrm{T}})^{-1}$ not to the scaled negative gradient direction but to the direction

$$P_C(p^{(k)} - \tfrac{1}{\gamma}D(D^{\mathrm{T}}p^{(k)} - f)) - p^{(k)}. \tag{3.46}$$

Note that the new direction (3.46) is obtained by subtracting the last iterate $p^{(k)}$ from the point we get after performing a step of the forward-backward splitting or gradient descent reprojection algorithm. It has been reported in [120], see also Section 3.5, that the alternating split Bregman method needs fewer iterations than, e.g., the forward-backward splitting algorithm. This result can now be motivated since we have shown the close relation of the alternating split Bregman algorithm to a special Newton method, i.e., a *second-order method*.

Remark 3.4.4 *A different way to use the Levenberg-Marquardt method for our problem would be to first perform a Levenberg-Marquardt step with respect to a parameter θ_k and then to project onto the feasible set, i.e.,*

$$p^{(k+1)} = P_C\big(p^{(k)} + (\theta_k I + DD^{T})^{-1}(D(D^{T}p^{(k)} - f))\big). \tag{3.47}$$

Algorithms of this form were studied in [133, 174].

3.5 Comparison of DRS and FBS with multistep methods

In this section, we want to compare the FBS and DRS/alternating split Bregman methods to *multistep methods* in terms of their speed of convergence. The multistep methods we consider here are first-order methods which make use of the information gained in preceding iterations to find a better descent direction. In Section 3.5.1 we describe two multistep methods which were recently applied to image processing problems. We present results concerning the convergence speed of these multistep methods compared to the DRS/alternating split Bregman and the forward-backward splitting algorithm in Section 3.5.2.

3.5.1 Multistep methods

One source of information that is not used by all the algorithm presented so far is the history of the preceding iterations. We follow here [176] and call algorithms which exploit this additional information *multistep methods*. For solving linear systems of equations, one popular multistep technique is the conjugate gradient method. However, the multistep idea has not been applied to image processing problems until recently.

As an introductory example, let us consider the simple heavy ball method, cf., [176, pp. 65], which can improve the gradient descent algorithm for minimizing a sufficiently smooth function $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}$ without any constraints:

$$u^{(k+1)} = u^{(k)} - \gamma \nabla \mathcal{F}(u^{(k)}) + \eta(u^{(k)} - u^{(k-1)}), \tag{3.48}$$

for parameters $\gamma, \eta > 0$ specified in [176, pp. 65]. In the same way as for the conjugate gradient algorithm, the idea of the heavy ball method is to prevent the "zigzag" motion of $(u^{(k)})_{k\in\mathbb{N}}$ when approaching a minimizer. In other words, the additional term in (3.48) acts as "inertia" and produces a smoother trajectory as illustrated in [176, Fig. 6].

Based on an algorithm of Nesterov in [163], two interesting multistep methods for constrained minimization problems were proposed recently: The generalized fast iterative shrinkage thresholding algorithm (FISTA) of [19, 20] is a projected version of the algorithm in [163]. The method which is now widely known as Nesterov's algorithm, cf., [165], is a modification of the algorithm in [163] including projections. These algorithms are suitable for the following class of problems

$$\hat{u} = \operatorname*{argmin}_{u\in C} \mathcal{F}(u) \tag{3.49}$$

with the properties that

- $\mathcal{F}$ is Lipschitz continuously differentiable,
- $C \subset \mathbb{R}^N$ is nonempty, bounded, closed and convex,
- a solution $\hat{u}$ of problem (3.49) exists.

Note that the boundedness of C is only necessary for Nesterov's algorithm. To define Nesterov's algorithm more easily, we will further assume that C is centered around zero although this condition is not necessary in general. In the following, we will denote the corresponding Lipschitz constant by L. The above class of problems is of interest for many problems in image processing. In particular, it covers the dual ROF problem (3.24). Another application of Nesterov's algorithm, e.g., is to multi-class image labeling in [140]

We will now state *Nesterov's algorithm* in the original form given in [165] but for computations it is preferable to rewrite it as in [8, 223].

Algorithm (Nesterov's algorithm)
Initialization: $v^{(0)} = f$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} u^{(k+1)} &:= P_C(v^{(k)} - \frac{1}{L}\nabla\mathcal{F}(v^{(k)})), \\ w^{(k+1)} &:= P_C(-\frac{1}{L}\sum_{i=1}^{k}\frac{i+1}{2}\nabla\mathcal{F}(u^{(i)})), \\ v^{(k+1)} &:= \frac{2}{k+3}w^{(k+1)} + \frac{k+1}{k+3}u^{(k+1)}. \end{aligned} \tag{3.50}$$

It is shown in [164] that for any solution $\hat{u}$ of (3.49) the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by the above algorithm satisfies the relation

$$\mathcal{F}(u^{(k)}) - \mathcal{F}(\hat{u}) \leq \frac{4L\|\hat{u}\|_2}{k(k+1)}. \tag{3.51}$$

This means that the approximation error measured in terms of the function value decreases with rate $\mathcal{O}(\frac{1}{k^2})$. Observe that so far convergence of the sequence $(u^{(k)})_{k\in\mathbb{N}}$ generated by Nesterov's algorithm has not been proved. Clearly, it is guaranteed if $\mathcal{F}$ is coercive.

Now let us define FISTA:

Algorithm (FISTA)
Initialization: $u^{(0)} = v^{(0)} = f$, $t_0 = 1$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} u^{(k+1)} &:= P_C(v^{(k)} - \frac{1}{L}\nabla\mathcal{F}(v^{(k)})), \\ t_{k+1} &:= \frac{1+\sqrt{1+4t_k^2}}{2}, \\ v^{(k+1)} &:= u^{(k+1)} + \frac{t_k - 1}{t_{k+1}}(u^{(k+1)} - u^{(k)}). \end{aligned} \tag{3.52}$$

The same convergence rate of $\mathcal{O}(\frac{1}{k^2})$ holds for the FISTA algorithm, see [19, 20]. More specifically, we have

$$\mathcal{F}(u^{(k)}) - \mathcal{F}(\hat{u}) \leq \frac{2L\|u^{(0)} - \hat{u}\|_2}{(k+1)^2}. \tag{3.53}$$

Observe that the constants on the right-hand side of (3.51) and (3.53) increase only linearly with the dimension of the space $\mathbb{R}^N$. As stated before, the multistep algorithm in [163] is already the unconstrained version of FISTA, i.e., it only lacks the projection in the first step. It took 25 years until Beck and Teboulle picked up this idea and formulated and proved the FISTA algorithm in 2008. The authors also proposed a modification of FISTA, called *monotone* FISTA (MFISTA). This method updates $u^{(k+1)}$ in the first step of (3.52)

only if this leads to a decrease in the value of $\mathcal{F}$. However, for our denoising example of Subsection 3.5.2 MFISTA did not yield an improvement over FISTA.

Comparing Nesterov's algorithm and FISTA, we see that the first step of the two algorithms is the same, a gradient descent step with step length $1/L$ and a subsequent projection on C. Then, both algorithm seek a way to improve the gradient descent reprojection step using the information of the preceding step(s). In Nesterov's algorithm the (projected) weighted sum of *all* the preceding gradient descent direction is considered. FISTA, on the other hand, uses simply the difference $u^{(k+1)} - u^{(k)}$. In both cases, this improves the classical gradient descent reprojection or forward-backward splitting method. If, e.g., the gradient descent direction is the same for a number of iterations than both algorithm take larger steps into this direction. On the other hand, "zig-zagging" behavior near a minimizer is dampened since the new direction is a weighted average of preceding ones. Note that the information of only one preceding step is used in FISTA. Nesterov's algorithm takes all of the preceding gradient direction into account but they are weighted such that newer directions have a greater impact.

Another difference between the two algorithms is that for large k, the weighting factors in the last step of Nesterov's algorithm imply that we essentially perform gradient descent reprojection steps with step length $1/L$. For the FISTA algorithm, on the other hand $\frac{t_k - 1}{t_{k+1}} \to 1$ and consequently for large k the algorithm approximately takes a gradient descent reprojection step with step length $1/L$ to compute $u^{(k+1)}$ and then the next forward-backward splitting step is calculated at the point $2u^{(k+1)} - u^{(k)}$.

These multistep algorithms turn out to be very efficient for our denoising problems, cf., [8, 223] and our numerical experiments below. Moreover, their convergence rate

$$\mathcal{F}(u^{(k)}) - \mathcal{F}(\hat{u}) = \mathcal{O}\big(\frac{1}{k^2}\big)$$

is analytically proved to be *optimal* for problems of the form (3.49) when we restrict ourselves to *first-order* methods, see [162, 164]. We will see in the next Section 3.5.2 that for a given number of iterations the alternating split Bregman performs even better numerically with respect to the error in the function value. This is not a contradiction to the optimality of the multistep methods described above since in Section 3.4.4 we have characterized the alternating split Bregman method as a special Newton method.

In our opinion, it is worthwhile to study the multistep methods in more detail in the future. It is unclear if other implementations of the multistep idea can yield even better results. Moreover, it seems promising to combine this idea with other minimization algorithms, e.g., the alternating split Bregman method. Another question is the relation between multistep methods and gradient descent techniques with adaptive step sizes, cf., [146, 234]. If, for example, an adaptive strategy will increase the step length in a "good" descent direction, then a multistep strategy might also, after a couple of iterations, do longer steps in this direction since it gets the information from the preceding steps that the current descent direction is a "good" one.

3.5.2 Numerical experiments

In this section, we compare for the ROF denoising problem (3.12) the numerical performance of the forward-backward shrinkage algorithm (3.35), Nesterov's algorithm (3.50), FISTA (3.52) and the DRS/alternating split Bregman method (3.31). All the computations of this section were done on a dual core desktop (2.4 GHz processors, 3 GB memory) using MATLAB 7.6.0.

For our numerical experiments, we consider the noisy 256×256 image shown in the middle picture of Fig. 3.3 which was obtained by adding white Gaussian noise of standard deviation 25 to the original image on the left of Fig. 3.3. We restrict our attention to the forward difference discretization $D := D_2$ of the gradient defined in (3.18). The corresponding denoising result with regularization parameter $\lambda = 25$ is shown on the right-hand side of Fig. 3.3. It is well-known that $\|D\|_2 < \sqrt{8}$ and this bound is the best possible

Figure 3.3: Left: Original image, values in $[0, 255]$. Middle: Noisy version with additive white Gaussian noise of standard deviation 25. Right: Denoising result using the ROF model with regularization parameter $\lambda = 25$.

if $n \to \infty$, cf., [56] or Appendix B. Hence, we choose $L = 1/\sqrt{8}$ for an approximation of the Lipschitz constant in Nesterov's algorithm and FISTA. For the forward-backward splitting shrinkage algorithm (3.35) we must choose $0 < \gamma < 2/\|D\|_2^2$ and it turns out that the best results are generally obtained for γ close to the upper bound. So, we choose $\gamma = 0.249$ for our experiments. For the first step (3.28) of the DRS/alternating split Bregman algorithm, we invert the matrix by means of the cosine-II matrix, cf., Appendix B. Note that we apply the cosine transform via matrix multiplication. A speed-up using fast cosine transform algorithm might be possible.

In the MATLAB implementations of the above algorithms we always use matrices instead of vectors to represent our images. This is possible since all our computations are separable with respect to the Kronecker product, i.e., for an image $U \in \mathbb{R}^{n,n}$ with corresponding vector $u \in \mathbb{R}^N$ and any $A \in \mathbb{R}^{r,n}, B \in \mathbb{R}^{s,n}$ the matrix BUA^{T} coincides with $(A \otimes B)u$ after reshaping.

Step length parameter for the alternating split Bregman algorithm. We have shown in Section 3.4 that the alternating split Bregman algorithm converges for every step length $\gamma > 0$. However, it turns out that the numerical results depend heavily on the choice of the specific value.

In the following experiment whose outcomes are depicted in Fig. 3.4, we have optimized the parameter γ numerically in two ways: For the number of iterations ranging from 1 to 1000 we have, first, computed the best γ such that the value of the objective function of the *dual* problem (3.24) is the smallest for three regularization parameters $\lambda = 15, 25$ and 35, cf. the top left plot of Fig. 3.4. Second, we have repeated this experiment to minimize the objective function of the *primal* problem (3.12). The results are shown in the top right plot of Fig. 3.4. In both experiments we have optimized γ with respect to a resolution of 0.001.

We see that the optimal value of γ behaves roughly as the scaled reciprocal function of the number of iterations. Furthermore, our experiments show that the parameter γ is larger for higher values of the regularization parameter λ. There seems to be only a slight difference when we compare the best values for γ optimized with respect to the dual and the primal function value. This is illustrated in Fig. 3.4 (bottom), where the solid and dotted curves show the result for the best γ regarding dual and primal function, respectively. These experiments give rise to two question for further research:

- Is it possible to speed up the alternating split Bregman method by using an *adaptive* step length strategy?
- Can the influence of the number of iterations and of the regularization parameter on the optimal value of the step length be described analytically?

It should be noted here, that the results presented in this paragraph hold also true for other images.

Algorithm comparison. Let us now compare the performance of the different algorithms. Related experimental results can also be found in [8, 223]. We measure the quality of the restoration as above in terms of the primal and dual function value. Moreover, we separate between two cases here. First, we consider the function values for a fixed number of iterations. Since the computational time for a single iteration will be different for the algorithms we then study the case where the running time is fixed in advance.

In Fig. 3.5, we see the results for fixed iteration numbers. In all cases, the alternating split Bregman algorithm performs best except for a relatively high number of iterations in the dual case and $\lambda = 15$, where the FISTA algorithm is slightly better, cf., the top left plot of Fig. 3.5. This underlines again the fact, that the alternating split Bregman algorithms is essentially a second-order method. Furthermore, we see that Nesterov's algorithm and FISTA always outperform the classical gradient descent reprojection method with fixed step size. It might be interesting to see in the future how the multistep methods perform compared to the forward-backward splitting algorithm improved by adaptive step length strategies. Comparing FISTA and Nesterov's algorithm, we observe that the latter has

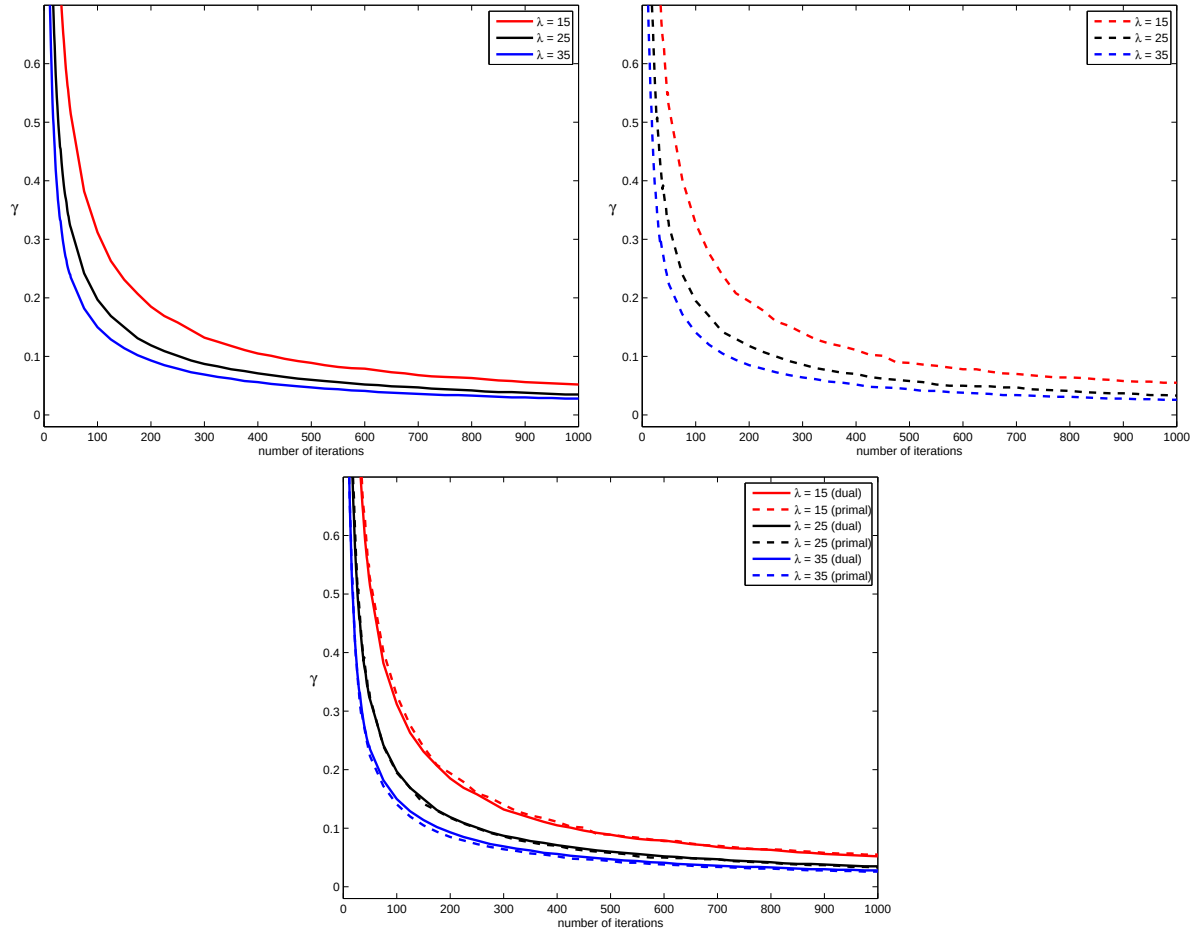


Figure 3.4: Top: Optimal values for the parameter γ in the alternating split Bregman algorithm optimized with respect to the dual objective function (left) and the primal functional (right) for $\lambda = 15, 25, 35$. Bottom: The overlay of the six curves from the top plots.

advantages regarding the primal functional while FISTA performs much better in terms of the value of the dual objective function. Note that these results hold true for other images and different image sizes.

Second, we consider speed comparisons for fixed running times. We restrict our attention here to the case of $\lambda = 25$. In Table 3.1, we see the results in terms of the logarithmic error in the *dual* objective function. Now, the FISTA algorithm outperforms the alternating split Bregman method since a single iteration of FISTA is much faster than one of the alternating split Bregman algorithm. The reason for this is that in the latter we have to multiply non-sparse matrices. Nesterov's algorithm is slightly slower in each iteration than FISTA so that it is clear that it performs worse than FISTA but it is still better than the alternating split Bregman algorithm. Although one iteration of the forward-backward splitting algorithm is calculate faster than one iteration of all the other algorithms, its poor results for a fixed numbers of iterations, cf., Fig. 3.5, carry over to this fixed-time comparison.

In Table 3.2, we consider the same experiment for the logarithmic error in the *primal* objective function. The differences to the preceding results are that now the alternating split Bregman algorithm shows the best performance and Nesterov's algorithm outperforms FISTA. The reason for the first observation is that for a fixed number of iterations the advantages of the alternating split Bregman algorithms are much greater than in the case of the dual functional, see Fig. 3.5. Second, Nesterov's algorithm is better than FISTA here, because it performs better for a fixed number of iterations which again outweighs the increased computation time per iteration.

It should be remarked here, that the results for the alternating split Bregman algorithm can be improved if we do not invert the matrix in the second step but use an approximation as done in [120].

Time	ASB	FISTA	Nesterov	FBS
5	4.62	2.73	3.28	7.15
10	2.45	0.10	1.28	5.80
15	1.33	-1.01	0.09	5.04
20	0.24	-2.05	-0.71	4.52
30	-0.92	-3.35	-1.62	3.93
40	-1.69	-4.47	-2.32	3.49
60	-3.33	-5.56	-3.68	2.68

Table 3.1: Logarithmic error with respect to the dual objective function for $\lambda = 25$ and different running times (in seconds).

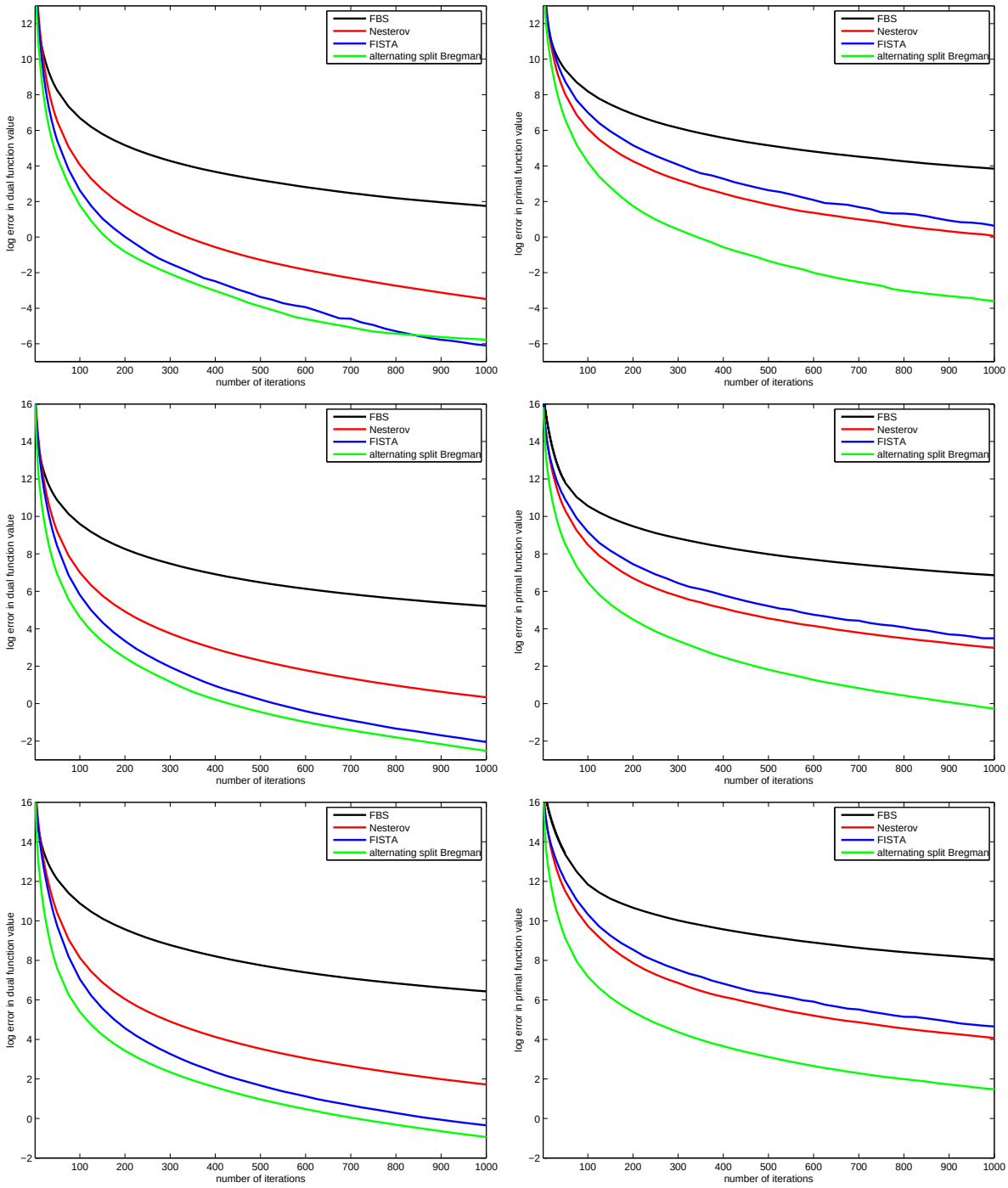


Figure 3.5: Logarithmic error in the dual function value (left) and the primal function value (right) for different algorithms for the ROF denoising model with regularization parameter $\lambda = 15$ (top), $\lambda = 25$ (middle) and $\lambda = 35$ (bottom).

Time	ASB	FISTA	Nesterov	FBS
5	6.47	7.02	5.38	8.56
10	4.50	5.14	3.73	7.40
15	3.48	4.25	2.78	6.71
20	2.48	3.32	2.14	6.26
30	1.27	2.28	1.36	5.73
40	0.40	1.52	0.73	5.34
60	-1.92	0.79	-0.47	4.61

Table 3.2: Logarithmic error with respect to the primal objective function for $\lambda = 25$ and different running times (in seconds).

3.6 Some generalizations of the ROF model

The ROF model (3.11) is well-known to preserve edges which is an important feature since edges often define objects in an image. It is natural, however, that this model has its limitations and must be adapted and generalized to yield good denoising results for a broader class of images.

A typical problem of the ROF model is, e.g., that it tends to introduce "staircases", i.e., constant areas where the image should be smooth. This is visible in Fig. 3.6 (middle left). One simple idea to prevent this is to use second-order derivatives. Let D_f again denote the forward difference matrix given by (3.16). We define the forward difference discretization of the second-order partial derivatives as

$$\mathcal{D}_2 = \begin{pmatrix} D_{xx} \\ D_{yy} \end{pmatrix} = \begin{pmatrix} I \otimes D_f^{\mathrm{T}} D_f \\ D_f^{\mathrm{T}} D_f \otimes I \end{pmatrix}, \tag{3.54}$$

other choices for the second-order difference matrix can be found in [90]. To avoid confusion, we denote the discrete gradient matrix D_2 from Section 3.3 as $D = \begin{pmatrix} D_x \\ D_y \end{pmatrix}$ here.

A typical result of solving (3.12) with the second-order difference matrix (3.54) is shown in the middle right picture of Fig. 3.6. The smooth areas are better restored and, in particular, the staircasing artifacts have disappeared. On the other hand, as expected, the edges are blurred now. In the following, we will briefly present two approaches to tackle both of these problems. For a more sophisticated discrete treatment of higher-order derivatives in ℓ_1 functionals, in particular, relations to contact problems and to discrete polynomial splines with higher defects, see [90] and also [201]. We also refer to [231] for a different higher-order method.

Note that we will need the second part of this section in Chapter 5.

Combined data and gradient fitting in conjunction with ℓ_1 regularization. We consider the model

$$\operatorname*{argmin}_{u\in\mathbb{R}^N} \frac{1}{2}\|u-f\|_2^2 + \frac{\beta}{2}\|Du-Df\|_2^2 + \lambda\||\mathcal{D}_2 u|\,\|_1, \tag{3.55}$$

which adds to a standard data-fitting and a second-order regularization term a new gradient fitting term. This term penalizes in the squared Euclidean norm the distance of the noisy and the restored gradient. In this way, we improve the restoration of sharp edges, cf. Fig. 3.6 (bottom left). Observe that

$$\frac{1}{2}\|u-f\|_2^2 + \frac{\beta}{2}\|Du-Df\|_2^2 = \frac{1}{2}\langle (I+\beta D^{\mathrm{T}}D)(u-f), u-f\rangle$$

and since $I+\beta D^{\mathrm{T}}D$ is positive definite there exists a matrix $B\in\mathbb{R}^{N,N}$ such that

$$B^{\mathrm{T}}B = I + \beta D^{\mathrm{T}}D.$$

So, a solution $\hat{u}$ of problem (3.55) can be computed as

$$\begin{aligned}\hat{u} &= \operatorname*{argmin}_{u\in\mathbb{R}^N} \frac{1}{2}\|B(u-f)\|_2^2 + \lambda\||\mathcal{D}_2 u|\,\|_1,\\ &= B^{-1}\operatorname*{argmin}_{v\in\mathbb{R}^N} \frac{1}{2}\|v-Bf\|_2^2 + \lambda\||\mathcal{D}_2 B^{-1}v|\,\|_1.\end{aligned} \tag{3.56}$$

In the same way as we derived the dual problem (3.26), we obtain the dual problem corresponding to (3.55) which is given by

$$\hat{b} = \operatorname*{argmin}_{b\in\mathbb{R}^{2N}} \frac{1}{2}\|B^{-\mathrm{T}}\mathcal{D}_2^{\mathrm{T}}b - Bf\|_2^2 \quad \text{s.t. } \|\,|b|\,\|_\infty \le \lambda \tag{3.57}$$

and the primal solution is recovered via $\hat{u} = f - B^{-1}B^{-\mathrm{T}}\mathcal{D}_2^{\mathrm{T}}\hat{b}$. Clearly, both the primal problem (3.56) and the dual problem (3.57) have the structure of problems (3.12) and (3.26), respectively. Hence, we can apply all the minimization techniques of this chapter. Observe that in order to compute the matrix B, we can use again the cosine transform decomposition presented in Appendix B, i.e.,

$$B = \left(C_{II}^{\mathrm{T}}\otimes C_{II}^{\mathrm{T}}\right)\left(I + \beta I_n\otimes \mathrm{diag}(q) + \beta\,\mathrm{diag}(q)\otimes I_n\right)^{\frac{1}{2}}\left(C_{II}\otimes C_{II}\right)$$

with $\mathrm{diag}(q) = (2\sin\frac{\pi j}{2n})^2$, $j=0,\dots,n-1$. Clearly, the square root of the diagonal matrix in the middle can be computed componentwise.

Infimal convolution regularization term. Let $g_1,\cdots,g_m$ be proper, convex and l.s.c. functions on $\mathbb{R}^N$, then the infimal convolution of these functionals is defined as

$$g_1\square\ldots\square g_m(u) = \inf_{u=u_1+\cdots+u_m}\{g_1(u_1)+\cdots+g_m(u_m)\}, \quad \forall u\in\mathbb{R}^N, \tag{3.58}$$

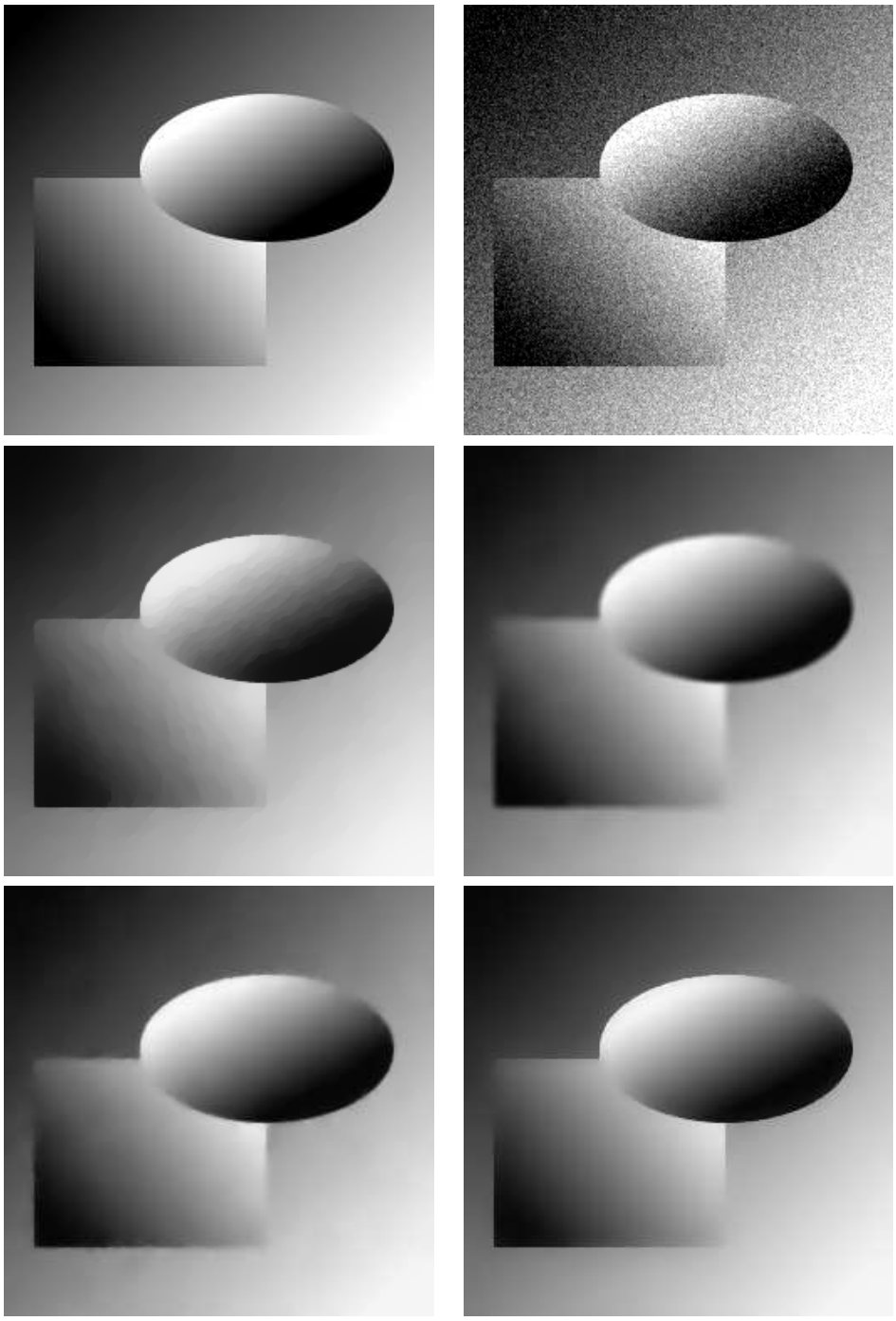


Figure 3.6: Top left: Original image (size 256×256, values in $[0, 255]$). Top right: Image with additive white Gaussian noise of standard deviation 20. Middle left: Denoised image with ROF model and $\lambda = 50$. The staircasing effect is visible. Middle right: Denoised image with second-order difference matrix (3.54) and $\lambda = 50$. The staircasing is reduced, but the edges are blurred. Bottom left: Denoised image via (3.55), $\beta = 0.8$, $\lambda = 50$. Edges become sharper. Bottom right: Denoised image via simplified inf-convolution model (3.62) and $\lambda_1 = 50$, $\lambda_2 = 180$.

cf., e.g., [181]. Observe that the envelope function (2.1) can be seen as the infimal convolution with the quadratic function $\frac{1}{2}\|\cdot\|_2^2$. In image processing an infimal convolution regularization term was first used in [59]. We consider here

$$\underset{u\in\mathbb{R}^N}{\operatorname{argmin}}\{\frac{1}{2}||u-f\|_2^2+(J_1\square J_2)(u)\}, \tag{3.59}$$

with

$$J_1(u):=\lambda_1\||Du|\|_1 \quad \text{and} \quad J_2(u):=\lambda_2\||\mathcal{D}_2u|\|_1,$$

cf. [62, 63, 90]. Clearly, problem (3.59) can be written as

$$\underset{u_1,u_2\in\mathbb{R}^N}{\operatorname{argmin}}\{\frac{1}{2}||u_1+u_2-f\|_2^2+\lambda_1\||Du_1|\|_1+\lambda_2\||\mathcal{D}_2u_2|\|_1\} \tag{3.60}$$

and the solution $\hat{u}$ of (3.59) is given by $\hat{u}=\hat{u}_1+\hat{u}_2$. It is well-known, cf., e.g., [181, Theorem 16.4] that the dual of the infimal convolution (3.58) is given by

$$(g_1\square\ldots\square g_2)^*=g_1^*+\cdots+g_m^*.$$

The functionals $g_1,\ldots,g_m$ do not even have to be l.s.c. for this statement to be true. Consequently, the dual problem to (3.59) has the form

$$\begin{aligned}\underset{b,b_1,b_2\in\mathbb{R}^{2N}}{\operatorname{argmin}}\ \{\frac{1}{2}\|b-f\|_2^2\} \quad \text{s.t.} \qquad & b=D^{\mathrm{T}}b_1=\mathcal{D}_2^{\mathrm{T}}b_2,\\ & \||b_1|\|_\infty\leq\lambda_1,\ \||b_2|\|_\infty\leq\lambda_2\end{aligned} \tag{3.61}$$

and $\hat{u}=f-\hat{b}$. For our special choice of the second-order difference matrix (3.54) it holds that

$$\mathcal{D}_2^{\mathrm{T}}=D^{\mathrm{T}}\begin{pmatrix}D_x & 0\\ 0 & D_y\end{pmatrix},$$

with $D_x=I\otimes D_f$ and $D_y=D_f\otimes I$. Assuming that in (3.61) it holds that

$$b_1=\begin{pmatrix}D_x & 0\\ 0 & D_y\end{pmatrix}b_2,$$

which is in general not true, we can modify (3.61) as follows:

$$\begin{aligned}\hat{b}_2=\underset{b_2\in\mathbb{R}^{2N}}{\operatorname{argmin}}\{\frac{1}{2}\|f-\mathcal{D}_2^{\mathrm{T}}b_2\|_2^2\} \quad \text{s.t.} \qquad & \|\,|\begin{pmatrix}D_x\\ D_y\end{pmatrix}b_2\,|\,\|_\infty\leq\lambda_1,\\ & \|\,|b_2|\,\|_\infty\leq\lambda_2.\end{aligned} \tag{3.62}$$

So, an approximation to the primal solution $\hat{u}$ is then obtained by $\tilde{u}=f-\mathcal{D}_2^{\mathrm{T}}\hat{b}_2$. The solution of the primal problem (3.60) as well as the dual problem (3.61) and its approximation (3.62) can be computed via second-order cone programming, cf., Chapter 5. The result of the approximate solution $\tilde{u}$ we get by solving (3.62) is very similar to the exact solution

$\hat{u}$ and the corresponding second-order cone algorithm converges faster than those for the two "exact" problems (3.60) and (3.61).

Alternatively, we can use an alternating minimization algorithm, fixing, e.g., in the primal problem alternatingly u_1 and u_2 and solving the corresponding ROF-like problems. The convergence of this procedure becomes clear by using the convergence results of averaged operators from Section 2.3.1.

However, this algorithm involves inner loops and one has to define corresponding stopping criteria. Very recently, T. Teuber [208] suggested to solve (3.60) directly via an alternating split Bregman method which does not contain inner loops. To this end, we apply the general alternating split Bregman algorithm (2.103)-(2.106) to the constrained problem

$$\begin{aligned} \underset{u_1,u_2,v_1,v_2\in\mathbb{R}^N,v_3,v_4\in\mathbb{R}^{2N}}{\operatorname{argmin}} & \{\underbrace{\frac{1}{2}||v_1+v_2-f||_2^2}_{=:g_1}+\underbrace{\lambda_1|||v_3|||_1}_{=:g_2}+\underbrace{\lambda_2|||v_4|||_1}_{=:g_3}\} \\ & \text{s.t.} \quad \begin{pmatrix} I & 0 \\ 0 & I \\ D & 0 \\ 0 & \mathcal{D}_2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix}. \end{aligned}$$

The hardest subproblems in the corresponding algorithm are the solution of linear systems with coefficient matrices $(I + D^{\mathrm{T}}D)$ and $(I + \mathcal{D}_2^{\mathrm{T}}\mathcal{D}_2)$ which can be again computed via the cosine-II transform or approximately via iterative methods for solving systems of linear equations.

A numerical result using the infimal convolution approach (3.59) is shown in Fig. 3.6 (bottom right). We see the remarkable performance of the infimal convolution method: the staircasing effect has disappeared without blurring the edges. The reason for this is that the infimal convolution model computes the restored image $\hat{u}$ as the sum of two components $\hat{u}_1$ and $\hat{u}_2$ which adapt themselves to the different regularization terms, cf., [203].

Remark 3.6.1 *Let us finally mention that many other useful regularization functionals are applied in the literature. Two popular choices are nonlocal and dictionary-based regularization terms. The idea of these relatively new classes of methods is to let the restoration algorithm itself compute the prior information about the space of images. For example, we can use the statistics of the given (noisy) image. This idea originated in* [41] *and the corresponding algorithms are often referred to as nonlocal methods. In a variational setting nonlocal methods were studied, e.g., in* [113, 114, 134] *and in connection with non-additive noise in* [203].

One can also use an appropriate dictionary to gain information about the image statistics. More on these dictionary-based approaches can be found, e.g., in [98, 149, 150].

3.7 Poisson noise removal

We can transfer the denoising model (3.12) to the case where the image is corrupted with Poisson noise by replacing the data-fitting term in (3.12) with a discrete version of (3.3). This leads to the following problem

$$\hat{u} = \underset{u\in\mathbb{R}^N}{\operatorname{argmin}}\{\sum_{i=1}^{N}\big(u_i - f_i\log(u_i)\big) + \Phi(Du)\}, \tag{3.63}$$

where the noisy image $f \in \mathbb{R}^N$ is assumed to be positive. Let us focus here on the total variation regularization term, i.e., $\Phi := \Phi_2$ and the gradient discretization $D := D_2$. This yields the problem

$$\hat{u} = \underset{u\in\mathbb{R}^N}{\operatorname{argmin}}\{\sum_{i=1}^{N}\big(u_i - f_i\log(u_i)\big) + \lambda\||Du|\|_1\}. \tag{3.64}$$

It should be noted here, that a similar functional was proposed for denoising in the presence of multiplicative Gamma noise in [199, 204]. We have also considered multiplicative methods in approximation theory in [198].

It turns out that gradient descent methods applied to the dual problem of (3.64) do not perform very well. Let us thus consider the DRS/alternating split Bregman algorithms. Applying the simplest alternating split Bregman algorithm (2.76)-(2.78) yields

$$u^{(k+1)} = \underset{u\in\mathbb{R}^N}{\operatorname{argmin}}\big\{\sum_{i=1}^{N}\big(u_i - f_i\log(u_i)\big) + \frac{1}{2\gamma}\|b^{(k)} + Du - d^{(k)}\|^2\big\}, \tag{3.65}$$

$$d^{(k+1)} = \underset{d\in\mathbb{R}^{2N}}{\operatorname{argmin}}\big\{\Phi(d) + \frac{1}{2\gamma}\|b^{(k)} + Du^{(k+1)} - d\|^2\big\}, \tag{3.66}$$

$$b^{(k+1)} = b^{(k)} + Du^{(k+1)} - d^{(k+1)}. \tag{3.67}$$

In the first step (3.65), we have to find the vector $u^{(k+1)} > 0$ which satisfies

$$0 = \sum_{i=1}^{N}\big(1 - \frac{f_i}{u_i^{(k+1)}}\big) + \frac{1}{\gamma}D^{\mathrm{T}}Du^{(k+1)} + \frac{1}{\gamma}D^{\mathrm{T}}b^{(k)} - \frac{1}{\gamma}D^{\mathrm{T}}d^{(k)}.$$

This problem has to be solved iteratively, see [204].

However, it is possible to use the multiple splitting idea presented in Section 2.4.4 to further decompose problem (3.64). We consider the following constrained problem equivalent to (3.64):

$$\underset{u,v\in\mathbb{R}^N,d\in\mathbb{R}^{2N}}{\operatorname{argmin}}\{\langle 0,u\rangle + \sum_{i=1}^{N}\big(v_i - f_i\log(v_i)\big) + \lambda\||d|\|_1\}, \quad \text{s.t. } \begin{pmatrix} I \\ D \end{pmatrix} u = \begin{pmatrix} v \\ d \end{pmatrix}. \tag{3.68}$$

According to (2.95)-(2.97) the corresponding alternating split Bregman algorithm has the form

$$u^{(k+1)} = \operatorname*{argmin}_{u\in\mathbb{R}^N} \{\frac{1}{2}\|b_1^{(k)} + u - v^{(k)}\|^2 + \frac{1}{2}\|b_2^{(k)} + Du - d^{(k)}\|^2\}, \tag{3.69}$$

$$v^{(k+1)} = \operatorname*{argmin}_{v\in\mathbb{R}^{2N}} \{\sum_{i=1}^{N} \big(v_i - f_i \log(v_i)\big) + \frac{1}{2\gamma}\|b_1^{(k)} + u^{(k+1)} - v\|^2\}, \tag{3.70}$$

$$d^{(k+1)} = \operatorname*{argmin}_{d\in\mathbb{R}^{2N}} \{\lambda\||d|\|_1 + \frac{1}{2\gamma}\|b_2^{(k)} + Du^{(k+1)} - d\|^2\}, \tag{3.71}$$

$$\begin{pmatrix} b_1^{(k+1)} \\ b_2^{(k+1)} \end{pmatrix} = \begin{pmatrix} b_1^{(k)} \\ b_2^{(k)} \end{pmatrix} + \begin{pmatrix} I \\ D \end{pmatrix} u^{(k+1)} - \begin{pmatrix} v^{(k+1)} \\ d^{(k+1)} \end{pmatrix}. \tag{3.72}$$

Now all of these subproblems can be solved explicitly: In the first step (3.69), we have to invert the matrix $I + D^{\mathrm{T}}D$ as in the Gaussian case and we can, e.g., again apply the diagonalization via the cosine-II matrix. In contrast to algorithm (3.65)-(3.67), we can now solve step (3.70) explicitly since the components of v are decoupled:

$$v^{(k+1)} = \frac{1}{2}\left(b_1^{(k)} + u^{(k+1)} - \gamma + \sqrt{\left(b_1^{(k)} + u^{(k+1)} - \gamma\right)^2 + 4\gamma f}\right),$$

cf., [106]. The third step (3.71) can simply be solved via coupled shrinkage, i.e.,

$$d^{(k+1)} = \widetilde{\mathcal{T}}_{\gamma\lambda I}(b_2^{(k)} + Du^{(k+1)}).$$

A different algorithm to solve (3.64) is the EM-TV algorithm considered in [39, 40, 188]. In [197], references to other known algorithms can be found.

The advantages of multiple splittings become even more apparent if the image is blurred prior the corruption by Poisson noise. Then, our minimization problem has the form

$$\hat{u} = \operatorname*{argmin}_{u\in\mathbb{R}^N, u\geq 0} \{\sum_{i=1}^{N} \big((Ku)_i - f_i \log((Ku)_i)\big) + \lambda\||Du|\|_1\}, \tag{3.73}$$

where $K \in \mathbb{R}^{N,N}$ is a blur matrix. Note that now we have to ensure explicitly that $u \geq 0$. In [106], the alternating split Bregman algorithm (2.103)-(2.106) was applied to the following constrained version of problem (3.73)

$$\operatorname*{argmin}_{x,u,v\in\mathbb{R}^N,} \{\langle 0, x\rangle + \sum_{i=1}^{N} \big(v_i - f_i \log(v_i)\big) + \lambda\||Du|\|_1\}, \quad \text{s.t. } \begin{pmatrix} K \\ I \end{pmatrix} x = \begin{pmatrix} v \\ u \end{pmatrix}.$$

The resulting PIDAL algorithm of [106], however, has two drawbacks. First, the nonnegativity constraint on u is not included and, second, the minimization with respect to u requires us to solve an ROF denoising problem in each step. So, we are confronted with

the problem of choosing an appropriate stopping criterion. In [197], on the other hand, we avoid both of these problems by applying (2.103)-(2.106) to the constrained problem

$$\operatorname*{argmin}_{x,u,v\in\mathbb{R}^N,d\in\mathbb{R}^M} \{\langle 0,x\rangle + \sum_{i=1}^{N} \big(v_i - f_i \log(v_i)\big) + \lambda\||d|\|_1 + \iota_{\geq 0}(u)\}, \quad \text{s.t.} \begin{pmatrix} K \\ D \\ I \end{pmatrix} x = \begin{pmatrix} v \\ d \\ u \end{pmatrix},$$

which results in the so-called PIDSplit+ algorithm.

Another important form of non-Gaussian noise is impulse or salt-and-pepper noise. It is well-known that L_1 data-fitting terms are useful for this kind of noise, cf., e.g., [60, 166, 167, 168, 177]. Again, the alternating split Bregman approach allows us to decompose these problems into subproblems which can be solved explicitly via soft and coupled shrinkage, cf., [101].

CHAPTER 4

Application to image inpainting

4.1 Introduction

In the preceding chapters, the focus of our discussion was on finding fast algorithms for general minimization problems in image processing (Chapter 2) and denoising problems (Chapter 3). In this chapter, which is based on [61], we describe a contribution to the modeling side of image processing by proposing novel wavelet-based inpainting algorithms. Applying ideas from anisotropic regularization and diffusion, our model can better handle degraded pixels at edges. We interpret the resulting algorithms as applications of the forward-backward splitting method which also ensures that convergence is guaranteed. Numerical examples illustrate the good performance of our algorithms.

First, let us give a brief description of the *inpainting problem*. It occurs when part of the data in an image is missing. The task then is to recover the missing regions from the observed (sometimes noisy) incomplete data. The mathematical model for the image inpainting problem reads as follows: We will again work in the discrete setting described in Section 3.3, i.e., two-dimensional images defined on $\{1,\dots,n\}\times\{1,\dots,n\}$ will be reshaped columnwise into a vectors $u \in \mathbb{R}^N$ with $N = n^2$. Let the nonempty set $\Upsilon \subset \{1,\dots,N\}$ be the region of the given pixels. Then the observed incomplete image f is

$$f_j = \begin{cases} u_j + \varepsilon_j & \text{if } j \in \Upsilon, \\ \text{arbitrary} & \text{otherwise,} \end{cases}$$

where $\varepsilon(j)$ denotes the noise. In the following, we denote by P_Υ the diagonal matrix with diagonal entries 1 for indices in Υ and 0 otherwise.

Initiated by [24], many useful techniques have been proposed to address this problem. In this chapter, we are mainly interested in wavelet-based inpainting methods. Such methods were, e.g., proposed in [51, 104]. However, these methods often let degraded pixels survive at sharp edges. A typical example is shown in Fig. 4.1. Here both the cubic spline interpolation and the wavelet-based method from [51] produce visible artifacts, in particular at the horizontal line. This was our motivation for considering more flexible wavelet-based methods.

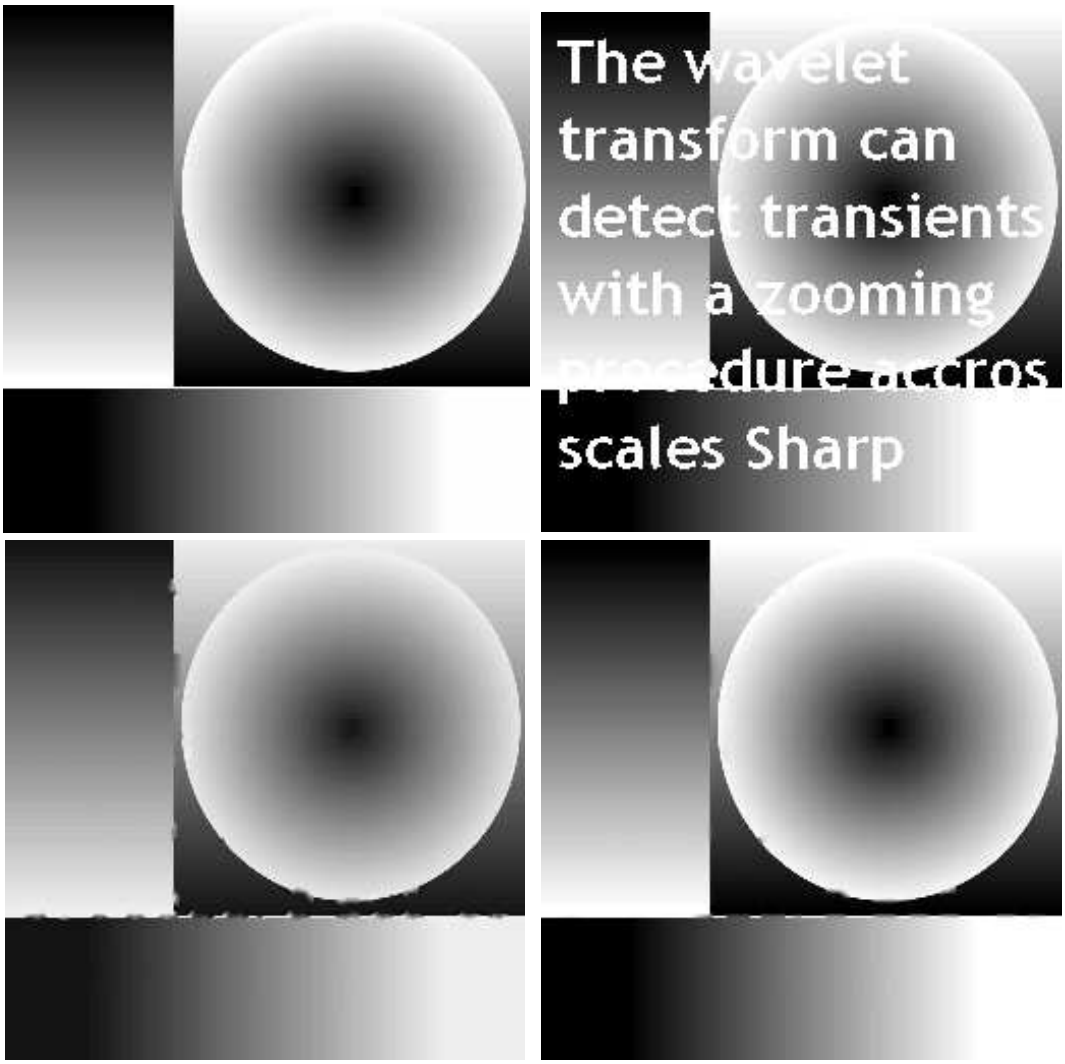


Figure 4.1: Top left: original image. Top right: degraded image. Bottom left: cubic interpolation by the MATLAB routine "griddata" (Peak signal-to-noise ratio (PSNR) = 29.39, $\text{err}_2 = 8.64$, $\text{err}_1 = 0.79$, cf., Section 4.5). Bottom right: interpolation by the algorithm in [51] with $c = 1$ and two levels (PSNR=33.27, $\text{err}_2 = 5.53$, $\text{err}_1 = 0.46$). The interpolated images have artifacts, e.g., at the horizontal line.

We focus on the following *general types of inpainting algorithms*.

Algorithm (Inpainting algorithm I - exact data)
Initialization: $u^{(0)} := f$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached
i) Solve a restoration problem for the current image $u^{(k)}$ to obtain $\hat{u}^{(k+1)}$.

ii) Set

$$u_j^{(k+1)} := \begin{cases} f_j & \text{if } j \in \Upsilon, \\ \hat{u}_j^{(k+1)} & \text{otherwise.} \end{cases}$$

Output: u^*

Algorithm (Inpainting algorithm II - noisy data)
Same as the general inpainting algorithm I except that we have to apply step i) to the final iterate u^* again.
Output: $u^\diamond = \hat{u}^*$

Indeed, depending on the restoration method used in step i), many known inpainting algorithms are of this general type. In [51], the following framelet based denoising method was proposed for step i) of the general inpainting algorithm I: Let $W \in \mathbb{R}^{M,N}$, $M \geq N$, denote a Parseval frame matrix, i.e., any $u \in \mathbb{R}^N$ can be written as $u = W^{\mathrm{T}} d$ and $W^{\mathrm{T}} W = I$. Further, let $\Lambda := \mathrm{diag}(\lambda)$ be a diagonal matrix containing the components of the vector $\lambda := (\lambda_j)_{j=1}^M$ as diagonal entries. Then the authors suggest solving the Besov-norm problem

$$d^{(k+1)} = \underset{d \in \mathbb{R}^M}{\mathrm{argmin}} \big\{ \frac{1}{2} \|W u^{(k)} - d\|_2^2 + \|\Lambda\, d\|_1 \big\} \tag{4.1}$$

via the fast soft shrinkage operation defined in (3.22). The solution $\mathcal{T}_\Lambda(W u^{(k)})$ is then transformed back to the image domain, i.e., we in total we get the following restoration step i)

$$\hat{u}^{(k+1)} = W^{\mathrm{T}} \mathcal{T}_\Lambda(W u^{(k)}). \tag{4.2}$$

In [51], it was proved that for noisy input data the iterates of the general inpainting algorithm II with restoration step (4.2) converge to $u^\diamond = W^{\mathrm{T}} \hat{d}$, where $\hat{d}$ is the solution of

$$\hat{d} = \underset{d \in \mathbb{R}^M}{\mathrm{argmin}} \big\{ \frac{1}{2} \|P_\Upsilon W^{\mathrm{T}} d - P_\Upsilon f\|_2^2 + \|\Lambda\, d\|_1 + \frac{1}{2} \|(I - W W^{\mathrm{T}}) d\|_2^2 \big\}. \tag{4.3}$$

Indeed this algorithm is very similar to a method proposed in [104], where the authors solve

$$\hat{d} = \underset{d \in \mathbb{R}^M}{\mathrm{argmin}} \big\{ \frac{1}{2} \|P_\Upsilon W^{\mathrm{T}} d - P_\Upsilon f\|_2^2 + \|\Lambda\, d\|_1 \big\} \tag{4.4}$$

by

$$d^{(k+1)} = \mathcal{T}_\Lambda \left(d^{(k)} + W(P_\Upsilon f - P_\Upsilon W^{\mathrm{T}} d^{(k)}) \right)$$

and set $u^\diamond = W^{\mathrm{T}} \hat{d}$. Obviously, for an orthogonal matrix W the wavelet-based algorithms (4.3) and (4.4) coincide. However, for various nonorthogonal frame analysis matrices W, the numerical experiments in [51] indicate that the algorithm (4.3) performs better. This can be explained by the fact that in (4.3) we not only focus on the sparsity of $\hat{d}$ but also introduce an additional regularity term.

In [99], the method (4.4) was generalized in order to recover both the texture and the cartoon part of an image, see also [25]. To this end, the authors solve

$$\operatorname*{argmin}_{d_t,d_c} \Big\{ \frac{1}{2} \|P_\Upsilon(W_t^{\mathrm{T}} d_t + W_c^{\mathrm{T}} d_c - f)\|_2^2 + \lambda(\|d_t\|_1 + \|d_c\|_1) + \gamma \| \, |(W_c^{\mathrm{T}} d_c)| \, \|_1 \Big\},$$

where W_c denotes the discrete curvelet transform, W_t denotes the discrete cosine transform, and d_t and d_n are the texture and cartoon components, respectively.

Beyond regularization techniques, PDE-based approaches can be applied in the restoration step. In [111, 225], it was demonstrated that inpainting methods based on edge enhancing anisotropic diffusion appear to be superior to linear methods, e.g., spline interpolation methods, and nonlinear isotropic diffusion methods. Indeed, these ideas were, together with wavelet techniques, the second ingredient for our algorithms. For other PDE-based methods, we refer only to [65] and the references therein.

In this chapter, we focus on inpainting by combining anisotropic regularization and diffusion methods with multilevel Haar-wavelet filters. Our new methods increase, compared to [51], the PSNR of various restored images significantly, e.g., by 3 dB for the image in Fig. 4.1, and avoid highly visible artifacts. Following the lines of [51], we prove the convergence of our method by embedding it into the framework of forward-backward splitting algorithms.

This chapter is organized as follows: In Section 4.2, we briefly review anisotropic regularization and diffusion methods. Ideas from this section, in particular the application of a diffusion tensor, carry over to our wavelet setting. In Section 4.3, we present new anisotropic Haar-wavelet methods for the inpainting problem. The convergence proof of our algorithms is given in Section 4.4. Finally, Section 4.5 contains numerical examples which demonstrate the excellent performance of our algorithms.

4.2 Anisotropic regularization and diffusion

In this section, we sketch the basic ideas from anisotropic diffusion and regularization methods that carry over to our wavelet setting. We prefer the more common continuous point of view in this section, while the rest of the chapter deals with a discrete setting obtained by discretizing gradients with the help of wavelet filters. Anisotropic diffusion methods such as edge enhancing or coherence enhancing diffusion have been used for the directed denoising of images for a long time, see [220] and the references therein. Recently, anisotropic regularization methods have become popular, e.g., for the restoration of polygonal shapes [22, 100, 196] with sharp edges and corners.

Let us consider a single restoration step r of our inpainting method which computes for a given continuous image $\tilde{f} := u^{(k)}$ on a quadratic domain Ω the image $\hat{u}^{(k+1)}$. By $\circ$, we denote the Hadamard product (componentwise product) of matrices. From the variational point of view, one could restore the image by solving for an appropriate proper, convex

and l.s.c. function Φ and an invertible matrix $V \in \mathbb{R}^{2,2}$ the problem

$$\operatorname*{argmin}_u \Big\{ \frac{1}{2}\|\tilde{u} - f\|_{L_2}^2 + \lambda \int_\Omega \Phi((V^{\mathrm{T}}\nabla u) \circ (V^{\mathrm{T}}\nabla u))\, dxdy \Big\}, \tag{4.5}$$

where the function space of u depends on the choice of Φ. For $\Phi(x^2, y^2) := \sqrt{x^2 + y^2}$ and $V := I$, the functional in (4.5) is the again the ROF functional defined on the space BV of functions of bounded variations, cf. (3.11). For $\Phi(x^2, y^2) := |x| + |y|$ and special rotation matrices V, the functional (4.5) was used for corner preserving denoising in [22, 196, 203]. For $V = I$, minimization algorithms for this functional were considered, e.g., in [129]. If Φ is differentiable, then the Euler–Lagrange equation of (4.5) reads

$$0 = \tilde{f} - u + \lambda \nabla \cdot (D\, \nabla u) \tag{4.6}$$

with

$$D := V \begin{pmatrix} 2\partial_1\Phi\big((V^{\mathrm{T}}\nabla u) \circ (V^{\mathrm{T}}\nabla u)\big) & 0 \\ 0 & 2\partial_2\Phi\big((V^{\mathrm{T}}\nabla u) \circ (V^{\mathrm{T}}\nabla u)\big) \end{pmatrix} V^{\mathrm{T}}. \tag{4.7}$$

Here, ∂_ν denotes the derivative with respect to the ν-th variable. For example, we have for the mapping $\Phi(x^2, y^2) := \sqrt{x^2 + y^2 + \varepsilon^2}$ that

$$\partial_1\Phi(x^2, y^2) = \partial_2\Phi(x^2, y^2) = 1/(2\sqrt{x^2 + y^2 + \varepsilon^2})$$

and for $\Phi(x^2, y^2) := \sqrt{x^2 + \varepsilon^2} + \sqrt{y^2 + \varepsilon^2}$ that

$$\partial_1\Phi(x^2, y^2) = 1/(2\sqrt{x^2 + \varepsilon^2})$$

and

$$\partial_2\Phi(x^2, y^2) = 1/(2\sqrt{y^2 + \varepsilon^2}).$$

On the other hand, the so-called anisotropic *edge enhancing diffusion* (EED) acts via

$$\begin{aligned} \partial_t u &= \nabla \cdot (D\nabla u), \\ u(x, 0) &= \tilde{f}(x), \end{aligned} \tag{4.8}$$

with appropriate boundary conditions, mostly Neumann boundary conditions, to prevent the loss of energy and with the *diffusion tensor*

$$D := V \begin{pmatrix} g(|\nabla u_\sigma|) & 0 \\ 0 & 1 \end{pmatrix} V^{\mathrm{T}}, \qquad V := (v \;\; v^\perp), \qquad v := \frac{\nabla u_\sigma}{|\nabla u_\sigma|}. \tag{4.9}$$

Here $u_\sigma = u * K_\sigma$ denotes the convolution of u with the Gaussian of standard deviation σ and g is a decreasing nonnegative function. In applications, the function

$$g(|s|) = \begin{cases} 1 - e^{-\frac{3.31488}{(s/\alpha)^8}} & s > 0, \\ 1 & s = 0, \end{cases}$$

introduced by Weickert in [220], has shown a good performance.

A relation to regularization methods can be seen as follows: If we use instead of (4.9) matrices of the form (4.7), then (4.6) can be considered as a semidiscretization of (4.8) via an implicit Euler step of time step size λ using these matrices. The following wavelet methods are related to explicit time discretizations so that we can achieve only approximations of the corresponding regularization method. For further investigations in this direction, see [191]. Note that according to [220], we will call a method *anisotropic* if the diagonal matrix in the diffusion tensor contains different nonzero diagonal entries. In this sense, the ROF method is an isotropic one.

4.3 Anisotropic Haar-wavelet shrinkage

In this section, we return to our discrete setting. We show how the ideas of anisotropic regularization and diffusion presented in (4.5) and (4.8), respectively, can be used to replace problem (4.1) in the first step in the general inpainting algorithms I and II by more appropriate minimization problems without adding too much computational complexity to solve them.

First, let us consider the following discretization of the EED algorithm. Recall the Parseval frame

$$\begin{pmatrix} H_0 \otimes H_0 \\ H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix} \tag{4.10}$$

defined in (A.13) with respect to the filters of the Haar-wavelet $h_0 := \frac{1}{2}[1\ 1]$ and $h_1 := \frac{1}{2}[1\ -1]$. Note that for convenience of notation, $H_0 \in \mathbb{R}^{n,n}$ and $H_1 \in \mathbb{R}^{n,n}$ are again circulant matrices with respect to h_0 and h_1, i.e., we use periodic boundary conditions. A remark concerning Neumann boundary conditions can be found at the end of this section. Based on the frame (4.10), an ℓ_2-stable, conditionally consistent, so-called *locally semianalytic scheme* (LSAS), cf., [224], was developed for the numerical solution of the EED equation (4.8). It involves a sophisticated spatial discretization and an explicit Euler scheme as temporal discretization. For some starting image $u^{(0)}$ and a time step size τ the LSAS scheme computes at every time step the new image $u^{((j+1)\tau)}$ based on the preceding iterate $u^{(j\tau)}$ by the following steps:

1. $$\begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{pmatrix} := \begin{pmatrix} H_0 \otimes H_0 \\ H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix} u^{(j\tau)},$$

2. $$\begin{pmatrix} d_{01} \\ d_{10} \end{pmatrix} := V \begin{pmatrix} e^{-4\tau\, g(|\nabla u_\sigma^{(j\tau)}|)} & 0 \\ 0 & e^{-4\tau} \end{pmatrix} V^{\mathrm{T}} \begin{pmatrix} c_{01} \\ c_{10} \end{pmatrix},$$
$$\begin{aligned} d_{00} &:= c_{00}, \\ d_{11} &:= e^{-4\tau\,(g(|\nabla u_\sigma^{(j\tau)}|)+1)} c_{11}, \end{aligned}$$

3. $u^{((j+1)\tau)} := \begin{pmatrix} H_0 \otimes H_0 \\ H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix}^{\mathrm{T}} (d_{00}^{\mathrm{T}}, d_{01}^{\mathrm{T}}, d_{10}^{\mathrm{T}}, d_{11}^{\mathrm{T}})^{\mathrm{T}},$

where

$$\nabla u_\sigma^{(j\tau)} := 2 \begin{pmatrix} H_0 \otimes H_1 \\ H_1 \otimes H_0 \end{pmatrix} (u_\sigma^{(j\tau)} * K_\sigma)$$

and V is chosen in accordance with (4.9) as $V := \begin{pmatrix} c & -s \\ s & c \end{pmatrix}$ with

$$\begin{aligned} c &:= \operatorname{diag}\big(((H_0 \otimes H_1)u_\sigma^{(j\tau)})/w\big), \\ s &:= \operatorname{diag}\big(((H_1 \otimes H_0)u_\sigma^{(j\tau)})/w\big), \\ w &:= \sqrt{\big((H_0 \otimes H_1)u_\sigma^{(j\tau)}\big)^2 + \big((H_1 \otimes H_0)u_\sigma^{(j\tau)}\big)^2} \end{aligned}$$

and componentwise quotients $((H_0 \otimes H_1)u_\sigma^{(j\tau)})/w$ and squares $\big((H_0 \otimes H_1)u_\sigma^{(j\tau)}\big)^2$ of vectors.

We consider now the Haar Parseval frame with L decomposition levels which generalizes (4.10). For $l = 1, \dots, L$, let $H_\nu^{(l)} \in \mathbb{R}^{n,n}$, $\nu \in \{0, 1\}$, be the circulant matrix corresponding to the filter $h_\nu^{(l)} = \frac{1}{2}(1, \underbrace{0, \dots, 0}_{2^{l-1}-1}, (-1)^\nu)$ with $2^{l-1} - 1$ inserted zeros between the filter coefficients. Further, we set

$$\begin{pmatrix} H_{00}^{(l)} \\ H^{(l)} \\ H_{11}^{(l)} \end{pmatrix} := \begin{pmatrix} H_{00}^{(l)} \\ H_{10}^{(l)} \\ H_{01}^{(l)} \\ H_{11}^{(l)} \end{pmatrix} = \begin{pmatrix} H_0^{(l)} \otimes H_0^{(l)} \\ H_0^{(l)} \otimes H_1^{(l)} \\ H_1^{(l)} \otimes H_0^{(l)} \\ H_1^{(l)} \otimes H_1^{(l)} \end{pmatrix} \prod_{i=1}^{l-1} (H_0^{(l-i)} \otimes H_0^{(l-i)}).$$

Then the matrix

$$W := \begin{pmatrix} H_{00}^{(L)} \\ H^{(1)} \\ \vdots \\ H^{(L)} \\ H_{11}^{(1)} \\ \vdots \\ H_{11}^{(L)} \end{pmatrix} \in \mathbb{R}^{(3L+1)N, N} \tag{4.11}$$

satisfies $W^{\mathrm{T}}W = I$ while $WW^{\mathrm{T}} \neq I$. Let $V^{(k)}$ be orthogonal matrices and let

$$\Lambda^{(k)} := \operatorname{diag}\left(\lambda_j^{(k)}\right)_{j=1}^{2N}, \quad \Lambda_{11}^{(k)} := \operatorname{diag}\left(\lambda_{11,j}^{(k)}\right)_{j=1}^{N}, \quad k = 1, \dots, L,$$

be diagonal matrices with nonnegative entries. For $p \in [1,2]$, we consider the minimization problem

$$\operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{ \frac{1}{2}\|u-f\|_2^2 + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda^{(k)}\big(V^{(k)}\big)^{\mathrm{T}} H^{(k)} u\|_p^p + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda_{11}^{(k)} H_{11}^{(k)} u\|_p^p \big\}.$$

In our numerical examples, we will use only $p = 1$ and $p = 2$. Since $W^{\mathrm{T}}W = I$, this is equivalent to

$$\operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{ \frac{1}{2}\|Wu-Wf\|_2^2 + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda^{(k)}\big(V^{(k)}\big)^{\mathrm{T}} H^{(k)} u\|_p^p + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda_{11}^{(k)} H_{11}^{(k)} u\|_p^p \big\}.$$

Using the notation

$$c := Wf = \Big(c_{00}^{(L)},\ c^{(1)},\dots,c^{(L)},\ c_{11}^{(1)},\dots,c_{11}^{(L)}\Big)^{\mathrm{T}}, \qquad d := Wu,$$

this can be rewritten as

$$\operatorname*{argmin}_{d\in\mathbb{R}^{(3L+1)N}} \big\{ \frac{1}{2}\|d-c\|_2^2 + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda^{(k)}\big(V^{(k)}\big)^{\mathrm{T}} d^{(k)}\|_p^p + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda_{11}^{(k)} d_{11}^{(k)}\|_p^p \big\} \quad \text{s.t. } d \in \mathcal{R}(W).$$

Note that $d \in \mathcal{R}(W)$ is equivalent to $(I - WW^{\mathrm{T}})d = 0$, i.e., the orthogonal projection of d onto the kernel of W^{T} has to be 0. In other words, if $\hat{d}$ is a solution of this problem, then $WW^{\mathrm{T}}\hat{d}$ is just the orthogonal projection of $\hat{d}$ onto $\mathcal{R}(W)$. We will not solve this minimization problem in step i) of our inpainting algorithm, but rather the following problem which is obtained by neglecting the constraint:

$$\operatorname*{argmin}_{d\in\mathbb{R}^{(3L+1)N}} \big\{ \frac{1}{2}\|d-c\|_2^2 + J_{\Lambda,p}(d) \big\},$$

where

$$J_{\Lambda,p}(d) := \frac{1}{p}\sum_{k=1}^{L} \|\Lambda^{(k)}\big(V^{(k)}\big)^{\mathrm{T}} d^{(k)}\|_p^p + \frac{1}{p}\sum_{k=1}^{L} \|\Lambda_{11}^{(k)} d_{11}^{(k)}\|_p^p. \tag{4.12}$$

This functional can be decoupled as

$$\begin{aligned} &\frac{1}{2}\|d_{00}^{(L)} - c_{00}^{(L)}\|_2^2 \ + \ \sum_{k=1}^{L} \left(\frac{1}{2}\|d^{(k)} - c^{(k)}\|_2^2 + \frac{1}{p}\|\Lambda^{(k)}\big(V^{(k)}\big)^{\mathrm{T}} d^{(k)}\|_p^p \right) \\ &+ \ \sum_{k=1}^{L} \left(\frac{1}{2}\|d_{11}^{(k)} - c_{11}^{(k)}\|_2^2 + \frac{1}{p}\|\Lambda_{11}^{(k)} d_{11}^{(k)}\|_p^p \right). \end{aligned} \tag{4.13}$$

Now the three parts of the functional can be minimized separately, which leads to the following solution.

Lemma 4.3.1 *The minimizer $\hat{d}$ of the functional* (4.13) *is given by*

$$\begin{aligned} \hat{d}_{00}^{(L)} &= c_{00}^{(L)}, \\ \hat{d}^{(k)} &= V^{(k)}\, \mathcal{T}_{\Lambda^{(k)},p}\left((V^{(k)})^{\mathrm{T}} c^{(k)}\right), \quad k = 1, \dots, L, \\ \hat{d}_{11}^{(k)} &= \mathcal{T}_{\Lambda_{11}^{(k)},p}(c_{11}^{(k)}), \quad k = 1, \dots, L, \end{aligned} \tag{4.14}$$

with the following shrinkage procedures $\mathcal{T}_{\cdot,\cdot}$:

i) *the soft shrinkage $\mathcal{T}_{\Lambda,1} := \mathcal{T}_{\Lambda}$ for $p = 1$,*

ii) *$\mathcal{T}_{\Lambda,p}(y) = F_{\Lambda,p}^{-1}(y)$, where $F_{\Lambda,p}$ is the injective mapping*

$$F_{\Lambda,p}(x) = x + \Lambda^p(\mathrm{sgn}(x) \circ |x|^{p-1})$$

for $p \in (1, 2)$,

iii) *$\mathcal{T}_{\Lambda,2}(y) := (I + \Lambda^2)^{-1} y$ for $p = 2$.*

Moreover, we have for $p \in (1, 2]$ that

$$|\mathcal{T}_{\Lambda,p}(y)|^p \geq (I + \Lambda^p)^{-p}|y|^p - 1. \tag{4.15}$$

Proof: Since the matrices $V^{(k)}$ are orthogonal, we immediately obtain assertion i).

In the following, we restrict our attention to the central functional, i.e., to $\hat{d}^{(k)}$. For $p \in (1, 2]$, the functional is differentiable, and the minimizer has to fulfill

$$\begin{aligned} 0 &= \hat{d}^{(k)} - c^{(k)} + V^{(k)}\Lambda^{(k)}\Big(\mathrm{sgn}(\Lambda^{(k)}(V^{(k)})^{\mathrm{T}}\hat{d}^{(k)}) \circ \Big|\Lambda^{(k)}(V^{(k)})^{\mathrm{T}}\hat{d}^{(k)}\Big|^{p-1}\Big) \\ (V^{(k)})^{\mathrm{T}}c^{(k)}, &= (V^{(k)})^{\mathrm{T}}\hat{d}^{(k)} + \left(\Lambda^{(k)}\right)^p \mathrm{sgn}((V^{(k)})^{\mathrm{T}}\hat{d}^{(k)}) \circ |(V^{(k)})^{\mathrm{T}}\hat{d}^{(k)}|^{p-1}. \end{aligned}$$

Then $x = (V^{(k)})^{\mathrm{T}}\hat{d}^{(k)}$ is the solution of $(V^{(k)})^{\mathrm{T}}c^{(k)} = F_{\Lambda^{(k)},p}(x)$ and $\hat{d}^{(k)} = V^{(k)}x$. In particular, we have for $p = 2$ that $x = (I + \Lambda^2)^{-1}(V^{(k)})^{\mathrm{T}}c^{(k)}$.

We prove the last assertion (4.15) componentwise. For $x, y \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{\geq 0}$ the equation $y = x + \lambda^p \mathrm{sgn}(x)|x|^{p-1}$ implies that

$$|y| = |x| + \lambda^p |x|^{p-1}.$$

Then, we see for $|x| \geq 1$ and $p \in (1, 2]$ that $|y| \leq |x| + \lambda^p|x|$ and, consequently, $|x| \geq (1 + \lambda^p)^{-1}|y|$. For $|x| < 1$, we have that $|y| \leq |x|^{p-1} + \lambda^p|x|^{p-1}$ so that $1 > |x|^{p-1} \geq (1+\lambda^p)^{-1}|y|$. Thus, $1 > (1 + \lambda^p)^{-p}|y|^p$ and $|x|^p \geq 0 > (1 + \lambda^p)^{-p}|y|^p - 1$. □

Let us denote the whole shrinkage procedure by $\hat{d} = \mathcal{T}_{\Lambda,p}\, c$. Finally, we can compute the denoised image u of f by $u = W^{\mathrm{T}}\hat{d}$. With this denoising procedure our inpainting algorithm reads as follows.

Algorithm (Proposed inpainting algorithm I)
Initialization: $u^{(0)} := f$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached
i) Compute $\hat{u}^{(k+1)} = W^{\mathrm{T}}\mathcal{T}_{\Lambda,p}(Wu^{(k)})$ with $\mathcal{T}_{\Lambda,p}$ defined by Lemma 4.3.1.
ii) Set

$$u_j^{(k+1)} = \begin{cases} f_j & \text{if } j \in \Upsilon, \\ \hat{u}_j^{(k+1)} & \text{otherwise.} \end{cases}$$

Output: u^*

Algorithm (Proposed inpainting algorithm II)
Same as the proposed inpainting algorithm I except that we have to apply step i) to the final iterate u^* again.
Output: $u^\diamond = \hat{u}^*$

The set

$$C := \{g \in \mathbb{R}^N : g(j) = f(j),\ \forall j \in \Upsilon\}.$$

is nonempty, closed, and convex so that its indicator function ι_C is a proper l.s.c. convex function. Thus, step ii) of the inpainting procedure also reads

$$u^{(k+1)} = \operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{\frac{1}{2}\|u - \hat{u}^{(k+1)}\|_2^2 + \iota_C(u)\big\}.$$

Thus, the whole algorithm can be rewritten in the form

$$d^{(k+1)} = \operatorname*{argmin}_{d\in\mathbb{R}^{(3L+1)N}} \big\{\frac{1}{2}\|d - Wu^{(k)}\|_2^2 + J_{\Lambda,p}(d)\big\}, \tag{4.16}$$

$$u^{(k+1)} = \operatorname*{argmin}_{u\in\mathbb{R}^N} \big\{\frac{1}{2}\|u - W^{\mathrm{T}}d^{(k+1)}\|_2^2 + \iota_C(u)\big\}, \tag{4.17}$$

where $J_{\Lambda,p}(d)$ is defined in (4.12).

Remark 4.3.2 *(Neumann boundary conditions)*
If we assume mirrored boundary conditions, we have to replace the circulant matrices by Toeplitz matrices. We consider again the Haar filters h_i, $i = 1, 2$. Instead of circulant matrices, we use these now to construct Toeplitz matrices $H_i \in \mathbb{R}^{n+1,n+2}$, $i = 0, 1$. For the sake of simplicity, let us restrict our attention to a single decomposition level. Using the Toeplitz matrices H_1 and H_2, we again form the matrix

$$W = \begin{pmatrix} H_0 \otimes H_0 \\ H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix}. \tag{4.18}$$

For more than one decomposition level we would obtain a $(3L+1)(n+1)^2 \times (n+2)^2$ *matrix according to* (4.11)*. The matrix* W *of* (4.18) *is not a Parseval frame but using the fact that the Kronecker product is bilinear we obtain*

$$W^T W = \left(H_0^T H_0 + H_1^T H_1\right) \otimes \left(H_0^T H_0 + H_1^T H_1\right). \tag{4.19}$$

and since

$$H_0^T H_0 + H_1^T H_1 = \begin{pmatrix} \frac{1}{2} & & \\ & I & \\ & & \frac{1}{2} \end{pmatrix} \tag{4.20}$$

the Parseval frame property $W^T W = I$ *is almost fulfilled, except at boundary pixels.*

Let $\tilde{\mathbf{f}}$ *denote the image obtained from* $\mathbf{f}$ *by mirroring the boundaries by one pixel. Then, it follows from* (4.20) *and* (4.19) *that, after removing the auxiliary one-pixel boundary, the image* $W^T W \tilde{\mathbf{f}}$ *corresponds to* f *again.*

This procedure can be facilitated in the following way. Let $\tilde{H}_i \in \mathbb{R}^{n,n+1}$ *denote the matrices obtained from the Toeplitz matrices* H_i^T *by canceling their first and last rows. Then, we have that*

$$\tilde{H}_0 H_0 + \tilde{H}_1 H_1 = \begin{pmatrix} 0 & 1 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1 & 0 \end{pmatrix}.$$

So, let us define $\widetilde{W}$ *as described above for these matrices* $\tilde{H}_i$, $i = 1, 2$*. Since*

$$\widetilde{W} W = \left(\tilde{H}_0 H_0 + \tilde{H}_1 H_1\right) \otimes \left(\tilde{H}_0 H_0 + \tilde{H}_1 H_1\right)$$

it follows that $\widetilde{W} W \tilde{f} = f$*. Hence, applying Neumann boundary condition means that instead of* (4.16) *we solve in each iteration the minimization problem*

$$d^{(k+1)} = \underset{d \in \mathbb{R}^{4(n+1)^2}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|d - W \tilde{u}^{(k)}\|_2^2 + \frac{1}{p} \|\Lambda V^T d\|_p^p + \frac{1}{p} \|\Lambda_{11} d_{11}\|_p^p \right\}$$

with W *as defined in* (4.18) *and set* $\hat{u}^{(k+1)} = \tilde{W} d^{(k+1)}$*.*

4.4 Convergence considerations

Following [51], we show the convergence of our inpainting algorithm by identifying it as a forward-backward splitting algorithm to minimize the sum of two operators.

Theorem 4.4.1 *For every starting image* $u^{(0)}$ *and any* $p \in [1, 2]$ *the proposed inpainting algorithm I converges to a solution of the problem*

$$\underset{u \in \mathbb{R}^N}{\operatorname{argmin}} \mathcal{F}(u) \tag{4.21}$$

with

$$\mathcal{F}(u) = \min_{d\in\mathbb{R}^{(3L+1)N}} \{\frac{1}{2}\|d - Wu\|_2^2 + J_{\Lambda,p}(d)\} + \iota_C(u).$$

Furthermore, the proposed inpainting algorithm I can be identified as a forward-backward splitting method applied to this problem.

For the proof of Theorem 4.4.1 we will need the following lemmas.

Lemma 4.4.2 *The function $\mathcal{F}$ in* (4.21) *is coercive.*

Proof: Recall that in view of Lemma 4.3.1 we denote the unique minimizer of the first term of $\mathcal{F}$ by $\mathcal{T}_{\Lambda,p}(Wu)$ so that we have

$$\begin{aligned}\mathcal{F}(u) &= \frac{1}{2}\|Wu - \mathcal{T}_{\Lambda,p}(Wu)\|_2^2 + J_{\Lambda,p}(\mathcal{T}_{\Lambda,p}(Wu)) + \iota_C(u)\\ &\geq J_{\Lambda,p}(\mathcal{T}_{\Lambda,p}(Wu)).\end{aligned}$$

Let $Wu := \Big((Wu)_{00}^{(L)},\, (Wu)^{(1)},\ldots,(Wu)^{(L)},\, (Wu)_{11}^{(1)},\ldots,(Wu)_{11}^{(L)}\Big)^{\mathrm{T}}$. Then we see by (4.13) and (4.14) that

$$\begin{aligned}J_{\Lambda,p}(\mathcal{T}_{\Lambda,p}(Wu)) &= \frac{1}{p}\sum_{k=1}^{L}\|\Lambda^{(k)}\mathcal{T}_{\Lambda^{(k)},p}\big((V^{(k)})^{\mathrm{T}}(Wu)^{(k)}\big)\|_p^p\\ &\quad + \frac{1}{p}\sum_{k=1}^{L}\|\Lambda_{11}^{(k)}\mathcal{T}_{\Lambda_{11}^{(k)},p}((Wu)_{11}^{(k)})\|_p^p. \qquad (4.22)\end{aligned}$$

Now we have by (4.15) and by definition of the soft shrinkage function that

$$\frac{1}{p}\left(\lambda_j^{(k)}\right)^p\left|\mathcal{T}_{\lambda_j^{(k)},p}(y)\right|^p \geq \begin{cases}\lambda_j^{(k)}|y| - \left(\lambda_j^{(k)}\right)^2 & \text{for } p=1,\\ \frac{1}{p}\Big(\frac{\lambda_j^{(k)}}{1+\left(\lambda_j^{(k)}\right)^p}\Big)^p|y|^p - \frac{1}{p}\left(\lambda_j^{(k)}\right)^p & \text{for } p\in(1,2].\end{cases}$$

Thus, setting

$$\kappa_1 := \frac{1}{p}\min_{\substack{j=1,\ldots,2N\\ i=1,\ldots,N\\ k=1,\ldots,L}}\left\{\Big(\frac{\lambda_j^{(k)}}{1+\left(\lambda_j^{(k)}\right)^p}\Big)^p, \Big(\frac{\lambda_{11,i}^{(k)}}{1+\left(\lambda_{11,i}^{(k)}\right)^p}\Big)^p\right\}$$

and

$$\kappa_2 := \begin{cases}\sum_{k=1}^{L}\Big(\sum_{j=1}^{2N}\left(\lambda_j^{(k)}\right)^2 + \sum_{i=1}^{N}\left(\lambda_{11,j}^{(k)}\right)^2\Big) & \text{for } p=1,\\ \frac{1}{p}\sum_{k=1}^{L}\Big(\sum_{j=1}^{2N}\left(\lambda_j^{(k)}\right)^p + \sum_{i=1}^{N}\left(\lambda_{11,j}^{(k)}\right)^p\Big) & \text{for } p\in(1,2]\end{cases}$$

and applying that $\|x\|_p \geq \|x\|_2$ for $p \in [1,2]$, we get

$$\begin{aligned} J_{\Lambda,p}(\mathcal{T}_{\Lambda,p}(Wu)) &\geq \kappa_1\Big(\sum_{k=1}^{L}\|(V^{(k)})^{\mathrm{T}}(Wu)^{(k)}\|_p^p + \sum_{k=1}^{L}\|(Wu)_{11}^{(k)}\|_p^p\Big) - \kappa_2 \\ &\geq \kappa_1\Big(\sum_{k=1}^{L}\|(Wu)^{(k)}\|_2^p + \sum_{k=1}^{L}\|(Wu)_{11}^{(k)}\|_2^p\Big) - \kappa_2. \end{aligned}$$

Using the notation $W_0 := H_{00}^{(L)}$ and

$$W_1 := \big((H^{(1)})^{\mathrm{T}}, \ldots, (H^{(L)})^{\mathrm{T}}, (H^{(1)})_{11}^{\mathrm{T}}, \ldots, (H^{(L)})_{11}^{\mathrm{T}}\big)^{\mathrm{T}},$$

this can be rewritten as

$$J_{\Lambda,p}(\mathcal{T}_{\Lambda,p}(Wu)) \geq \kappa_1\|W_1u\|_2^p - \kappa_2. \tag{4.23}$$

By Lemma 4.4.3 below, the matrix $W_0^{\mathrm{T}}W_0$ has the simple eigenvalue 1 with a corresponding normed eigenvector $\tilde{u} = \frac{1}{\sqrt{N}}\mathbf{1}_N$. Since $W^{\mathrm{T}}W = I$, it follows that $W_1^{\mathrm{T}}W_1$ has the simple eigenvalue 0 and that the kernel of $W_1^{\mathrm{T}}W_1$ is spanned by $\tilde{u}$. Now we obtain for the orthogonal decomposition $u = v + a\tilde{u}$ that $|a| \geq \|u\|_2 - \|v\|_2$ and

$$\|W_1u\|_2^2 = \|W_1v\|_2^2 \geq \eta_2\|v\|_2^2, \tag{4.24}$$

where $\eta_2 > 0$ is the second smallest eigenvalue of $W_1^{\mathrm{T}}W_1$. Now we fix a constant $c \in (\frac{1}{\sqrt{N}+1}, 1)$ and consider two cases:
1. For $\|v\|_2 \geq c\,\|u\|_2$, we conclude by (4.23) and (4.24) that

$$\mathcal{F}(u) \geq \kappa_1\|W_1u\|_2^p - \kappa_2 \geq \kappa_1\,\sqrt{\eta_2}^p\,\|v\|_2^p - \kappa_2 \geq \kappa_1\,\sqrt{\eta_2}^p\,c^p\,\|u\|_2^p - \kappa_2.$$

2. For $\|v\|_2 < c\|u\|_2$ it holds that $|a| > (1-c)\|u\|_2$. Hence, we have for any $i_0 \in \Upsilon$ that

$$|u_{i_0}| \;=\; |v_{i_0} + a\tilde{u}_{i_0}| \geq |a||\tilde{u}_{i_0}| - |v_{i_0}| > (1-c)\|u\|_2|\tilde{u}_{i_0}| - c\|u\|_2 = \|u\|_2\frac{1-c(1+\sqrt{N})}{\sqrt{N}}.$$

Thus, we see for $\|u\|_2$ large enough that $|u_{i_0}| > |f_{i_0}|$ and, consequently, $\mathcal{F}(u) \geq \iota_C(u) = +\infty$. □

Lemma 4.4.3 *The matrix $W_0^TW_0$ has 1 as a simple eigenvalue with corresponding eigenvector $\tilde{u} = \frac{1}{\sqrt{N}}\mathbf{1}_N$.*

Proof: Using multiplication rules for tensor products, we obtain that

$$W_0^{\mathrm{T}}W_0 = B^{\mathrm{T}}B \otimes B^{\mathrm{T}}B, \quad B := \prod_{l=1}^{L} H_0^{(l)} = \frac{1}{2^L}\mathrm{circ}\,([\underbrace{1\ldots1}_{2^L}\,0\ldots0]).$$

By [86], the circulant matrix B has eigenvectors $\frac{1}{\sqrt{n}}(e^{-\frac{2\pi ijk}{n}})_{j=0}^{n}$ and eigenvalues $\beta_0 = 1$ and

$$|\beta_k| = |\frac{1}{2^L}\sum_{j=0}^{2^L-1} e^{-\frac{2\pi ijk}{n}}| \;=\; \frac{1}{2^L}\frac{|1-e^{-\frac{2\pi ij2^L}{n}}|}{|1-e^{-\frac{2\pi ij}{n}}|} = \frac{1}{2^L}\prod_{p=1}^{L}|1+e^{-\frac{2\pi ij2^{L-p}}{n}}| < 1,$$

$k = 1, \ldots, n-1$. The last inequality holds true because $|1+e^{-\frac{2\pi ij2^{L-p}}{n}}| \leq 2$ for $p = 1, \ldots, L$ with strict inequality for $p = L$. □

Proof of Theorem 4.4.1: Observe that the first term in (4.21) can be written in terms of an envelope as

$$\left({}^1J_{\Lambda,p} \circ W\right)(u) = \min_{d\in\mathbb{R}^{(3L+1)N}} \{\frac{1}{2}\|d - Wu\|_2^2 + J_{\Lambda,p}(d)\}.$$

Both functionals ${}^1J_{\Lambda,p} \circ W$ and ι_C are proper, convex and l.s.c. The first one is also differentiable so that we can apply the forward-backward splitting algorithm (2.30) with $B = \nabla\left({}^1J_{\Lambda,p} \circ W\right)$ and $A = \partial\iota_C$. As we will see below, we want to use the step length $\gamma = 1$ in the forward-backward splitting algorithm. This is an admissible parameter since the gradient

$$\nabla\left({}^1J_{\Lambda,p} \circ W\right)(u) = W^{\mathrm{T}}\left(Wu - \mathrm{prox}_{J_{\Lambda,p}}(Wu)\right)$$

is Lipschitz continuous with Lipschitz constant one which can be shown as follows, cf. [51]: First, observe that W being a Parseval frame implies $\|W\|_2 = 1$. So, using that the proximal operator is firmly nonexpansive we can apply (2.19) to get for any $u, v \in \mathbb{R}^N$

$$\begin{aligned}
&\|\nabla\left({}^1J_{\Lambda,p} \circ W\right)(u) - \nabla\left({}^1J_{\Lambda,p} \circ W\right)(v)\|_2 \\
=\;& \|W^{\mathrm{T}}\left(Wu - \mathrm{prox}_{J_{\Lambda,p}}(Wu)\right) - W^{\mathrm{T}}\left(Wv - \mathrm{prox}_{J_{\Lambda,p}}(Wv)\right)\|_2 \\
\leq\;& \|W\|_2\,\|Wu - \mathrm{prox}_{J_{\Lambda,p}}(u) - \left(Wv - \mathrm{prox}_{J_{\Lambda,p}}(v)\right)\|_2 \\
\leq\;& \|W(u-v)\|_2 \leq \|u-v\|_2.
\end{aligned}$$

We have thus shown that $\nabla\left({}^1J_{\Lambda,p} \circ W\right)$ is Lipschitz continuous with parameter one. Hence, the forward-backward splitting algorithm with $\gamma = 1$ is given by

$$\begin{aligned}
u^{(k+1)} &= \mathrm{prox}_{\iota_C}\left(u^{(k)} - \nabla\left({}^1J_{\Lambda,p} \circ W\right)(u^{(k)})\right) \\
&= \operatorname*{argmin}_{u\in\mathbb{R}^N}\left\{\frac{1}{2}\|u^{(k)} - \nabla\left({}^1J_{\Lambda,p} \circ W\right)(u^{(k)}) - u\|_2^2 + \iota_C(u)\right\} \\
&= \operatorname*{argmin}_{u\in\mathbb{R}^N}\left\{\frac{1}{2}\|u^{(k)} - W^{\mathrm{T}}\left(Wu^{(k)} - \mathrm{prox}_{J_{\Lambda,p}}(Wu^{(k)})\right) - u\|_2^2 \;+\; \iota_C(u)\right\} \\
&= \operatorname*{argmin}_{u\in\mathbb{R}^N}\left\{\frac{1}{2}\|W^{\mathrm{T}}\mathrm{prox}_{J_{\Lambda,p}}(Wu^{(k)}) - u\|_2^2 \;+\; \iota_C(u)\right\}. \qquad (4.25)
\end{aligned}$$

By (4.16) and (4.17) this coincides with the sequence produced by our proposed inpainting algorithm I. We conclude convergence of our algorithm since by Lemma (4.4.2) a solution of problem (4.21) exists. □

Remark 4.4.4 *Numerical experiments indicate that the proposed inpainting algorithm I converges linearly. However, we have not proved this so far. In* [211, Proposition 1(d)], *Tseng gives a sufficient condition for linear convergence. Unfortunately, it cannot be applied here since neither* $\nabla\,({}^1J_{\Lambda,p} \circ W)$ *nor* $\partial\iota_C$ *is strongly monotone.*

Remark 4.4.5 *Clearly, we can also use Nesterov's algorithm or FISTA instead of the above forward-backward splitting algorithm and make use of the information of the preceding iterations, cf., Subsection 3.5.1.*

4.5 Numerical examples

Finally, we present some numerical examples, in particular, we compare our algorithms with the algorithm in [51] without thresholding of the smoothest coefficients. Since the results for noisy data with a small amount of noise are similar as those for exact data, we restrict our attention to exact input data.

All programs were written in MATLAB. We have always assumed Neumann boundary conditions and we have used the following stopping criterion for the iterations: $\|u^{(k+1)} - u^{(k)}\|_2/\|u^{(k+1)}\|_2 \leq 5 \cdot 10^{-5}$. We compare the weighted ℓ_1-error $\mathrm{err}_1 := \|u - f\|_1/N$, the weighted ℓ_2-error $\mathrm{err}_2 := \|u - f\|_2/\sqrt{N}$, and the PSNR $:= 20 \cdot \log_{10}(255/\mathrm{err}_2)$. The parameters were chosen with respect to the "best" PSNR.

We compare the following algorithms:

(A) The framelet-based algorithm in [51] which uses the multi-level undecimated linear B-spline framelets defined in Section A.2 and soft shrinkage of the high-pass coefficients at level k with the thresholds $c/\sqrt{2^k}$. Note that, originally, Neumann boundary condition were used in [51] but the difference to periodic boundary conditions is neglectable.

(B) Proposed inpainting algorithm I with our Haar-wavelet filters, $p = 1$, $V^{(k)} = I$, and soft shrinkage with respect to the thresholds $\lambda_j^{(k)} := \lambda/\sqrt{2^k}$ and $\lambda_{11,j}^{(k)} := \lambda_{11}/\sqrt{2^k}$ at level k.

(C) The same algorithm as in (B) except that we use matrices $V^{(k)}$ inspired by the LSAS discussed in Section 4.3: We convolve an appropriate guess $\tilde{f}$ of the original function with the Gaussian of standard derivation σ to obtain $\tilde{f}_\sigma$. Then, at level k, we set

$$V^{(k)} := \begin{pmatrix} c^{(k)} & -s^{(k)} \\ s^{(k)} & c^{(k)} \end{pmatrix}$$

with $c^{(k)} := \mathrm{diag}\big(H_{01}^{(k)} \tilde{f}_\sigma/w\big)$, $s^{(k)} := \mathrm{diag}\big(H_{10}^{(k)} \tilde{f}_\sigma/w\big)$, and $w^{(k)} := \sqrt{(H_{01}^{(k)} \tilde{f}_\sigma)^2 + (H_{10}^{(k)} \tilde{f}_\sigma)^2}$, i.e., we use the same matrices $V^{(k)}$ in each iteration.

(D) Proposed inpainting algorithm I with $p = 2$ and the following setting inspired by the LSAS for EED defined in Section 4.3: We define $V^{(k)}$ as in (C). In the shrinkage step we use

$$\begin{aligned} \big(I + (\Lambda^{(k)})^2\big)^{-1} &:= \begin{pmatrix} \operatorname{diag}\big(e^{-4\tau\, g(w^{(k)})}\big) & 0 \\ 0 & \operatorname{diag}\big(e^{-4\tau}\, 1_N\big) \end{pmatrix}, \\ \big(I + (\Lambda_{11}^{(k)})^2\big)^{-1} &:= \operatorname{diag}\big(e^{-4\tau\,(g(w^{(k)})+1)}\big) \end{aligned}$$

with the vector 1_N of N ones. We still use the same matrices $V^{(k)}, \Lambda^{(k)}$, and $\Lambda_{11}^{(k)}$ in each iteration.

(E) The same algorithm as in (D) except that we do not freeze $V^{(k)}$ and the shrinkage matrices at the beginning of the algorithm with respect to $\tilde{f}_\sigma$ but compute them in each iteration step k with respect to the preceding iterate $u^{(k-1)}$. Note that we have not proved the convergence for this algorithm. If we would work only with one level of Haar-wavelet decomposition $L = 1$, then the restoration step can be considered as one time step of an iterative EED scheme discretized by LSAS.

In our **first example** we start with the image at the top right of Fig. 4.1 which we also use as initial guess $u^{(0)}$. Alternatively, one could use the bottom left image in Fig. 4.1 generated by the MATLAB cubic interpolation procedure "griddata" as initial guess. This leads to qualitatively similar results but with a smaller number of iterations. We have used this cubic interpolation in algorithms (D) and (E) for $\tilde{f}$. Detailed results are given in the tables below. Here "iter" denotes the number of iterations. The corresponding images for the decomposition level 2 are depicted in Fig. 4.2 (algorithms (B)–(E)) and at the bottom right of Fig. 4.1 (algorithm (A)). The algorithms (B)–(E) perform much better than the algorithm (A). The PSNR improves by approximately 3 dB if we use algorithms (B)–(D) and by approximately 5 dB for algorithm (E), compare Table 4.1. The Algorithms (B)–(E) considerably reduce the artifacts at the horizontal line. However, the methods (B) and (C) introduce some errors at the boundary of the circle. These artifacts do not appear if we apply the algorithms (D) and (E). Note that, in general, the PSNR cannot be substantially improved by choosing a higher decomposition level than $L = 2$.

In our **second example** we interpolate the image on the right-hand side of Fig. 4.3. The cubic interpolation is depicted at the top left of Fig. 4.4 and contains artifacts at the windows on the left-hand side. The results for our algorithms with two decomposition levels are as follows:

- Algorithm (A) with $c = 1.0$: PSNR = 31.61, $\text{err}_2 = 6.69$, $\text{err}_1 = 1.36$,
- Algorithm (B) with $\lambda = 0.5$ and $\lambda_{11} = 8$: PSNR = 34.08, $\text{err}_2 = 5.03$, $\text{err}_1 = 0.93$,
- Algorithm (C) with $\sigma = 0.5$, $\lambda = 0.5$ and $\lambda_{11} = 8$: PSNR = 33.98, $\text{err}_2 = 5.09$, $\text{err}_1 = 0.97$,

Table 4.1: Top to bottom: Results of the inpainting algorithm (A)-(E) for the first example.

Level	c	PSNR	err_2	err_1	iter
4	1.0	32.93	5.49	0.54	307
3	1.0	33.29	5.51	0.48	307
2	1.0	**33.27**	5.53	0.46	358
1	1.6	32.50	6.04	0.50	461

Level	λ	λ_{11}	PSNR	err_2	err_1	iter
4	1	8	34.84	4.61	0.36	272
3	1	10	35.52	4.27	0.29	235
2	1	100	**36.42**	3.84	0.27	278
1	1	100	35.89	4.09	0.28	814

Level	σ	λ	λ_{11}	PSNR	err_2	err_1	iter
4	4	1	8	35.43	4.31	0.36	244
3	4	1	10	35.97	4.05	0.30	223
2	4	1	100	**36.60**	3.76	0.26	269
1	4	1	100	36.03	4.02	0.26	811

Level	σ	τ	α	PSNR	err_2	err_1	iter
4	4	1	2	35.19	4.43	0.47	73
3	4	1	2	36.08	4.00	0.36	79
2	4	1	2	**36.79**	3.68	0.29	106
1	4	1	2	36.83	3.83	0.28	208

Level	σ	τ	α	PSNR	err_2	err_1	iter
4	4	1	2	35.91	4.07	0.44	78
3	4	1	2	37.47	3.40	0.29	88
2	4	1	2	**38.58**	2.99	0.23	123
1	4	1	2	37.99	3.21	0.24	215

- Algorithm (D) with $\sigma = 1$, $\tau = 1$ and $\alpha = 2$: PSNR = 31.56, $\text{err}_2 = 6.73$, $\text{err}_1 = 1.27$,
- Algorithm (E) with $\sigma = 1$, $\tau = 1$ and $\alpha = 2$: PSNR = 31.36, $\text{err}_2 = 6.89$, $\text{err}_1 = 1.26$.

Algorithms (B) and (C) perform best since the edges in the original image are mostly horizontal or vertical. The PSNR is approximately 2 dB higher than in the other three algorithms. While the algorithms (A), (D) and (E) produce similar artifacts especially at the windows, these errors do not appear if we apply the algorithms (B) and (C). This is illustrated in Fig. 4.4 and in the zoomed images in Fig. 4.5.

In our **third example**, we consider the image at the top left of Fig. 4.6. For this image cubic interpolation yields very good results (PSNR = 33.62), see top right of Fig. 4.6. Starting with this image as an initial guess and using small parameters ($c = \lambda = \lambda_{11} = 0.05$), we can achieve a PSNRs of around 33.8 by applying algorithms (A)–(C). Visual differences to the image obtained by cubic interpolation are hard to find. For algorithms (D) and (E) with the original image as initial guess, two decomposition levels and parameters $\sigma = 1$, $\tau = 1$, and $\alpha = 10$, we obtain the PSNR = 34.25, $\text{err}_2 = 4.93$, $\text{err}_1 = 0.98$ after 86 iterations and the PSNR = 34.21, $\text{err}_2 = 4.96$, $\text{err}_1 = 1.00$ after 249 iterations, respectively. As shown at the bottom of Fig. 4.6, there are visual differences at long edges.

Remark 4.5.1 *Finally, let us note that inpainting in both the image domain and the domain of the wavelet coefficients as well as applications to superresolution were considered in* [50, 49].

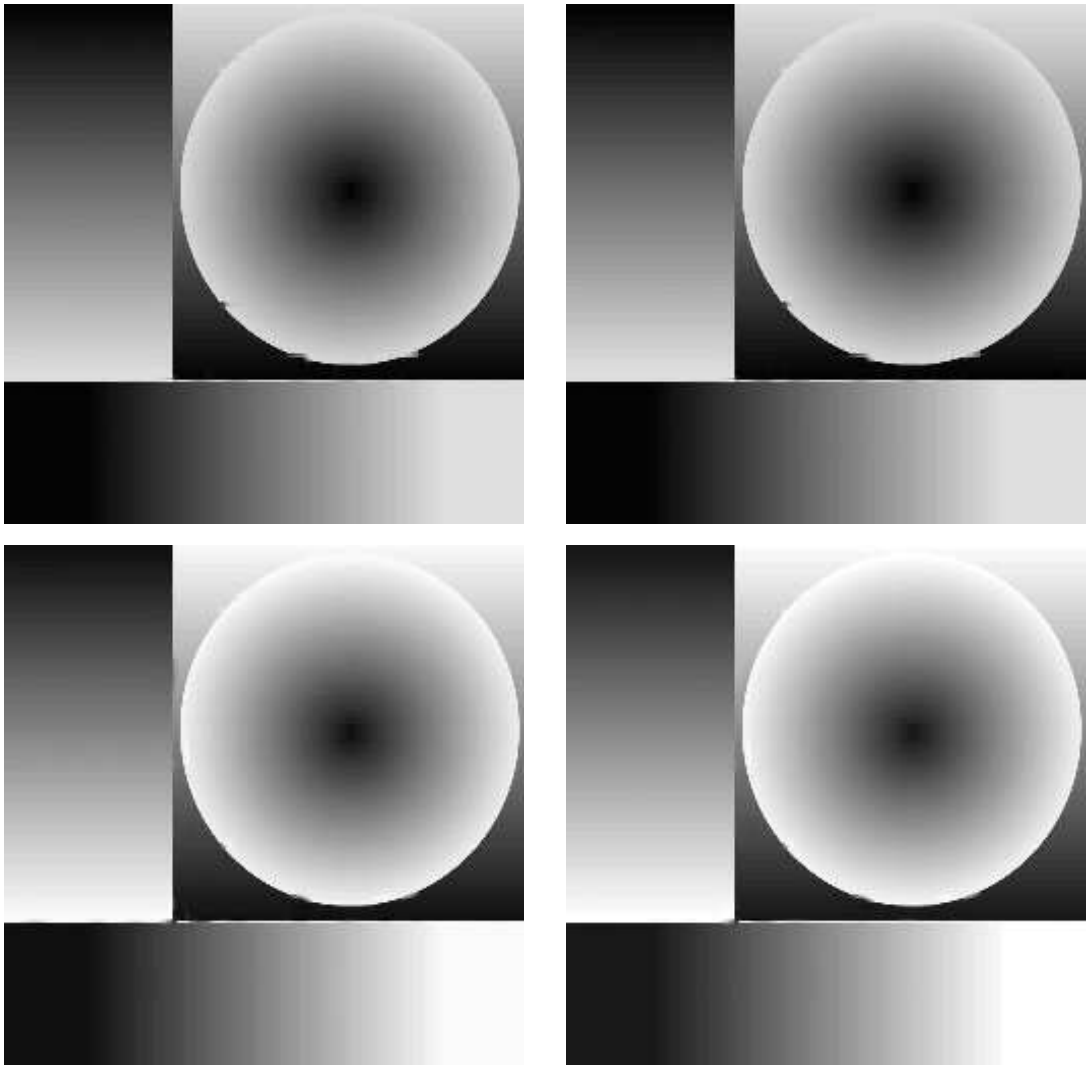


Figure 4.2: Applications of the algorithms (B)–(E) with decomposition level 2. Top left: Algorithm (B) (PSNR=36.42, $\mathrm{err}_2 = 3.84$, $\mathrm{err}_1 = 0.27$). Top right: Algorithm (C) (PSNR=36.60, $\mathrm{err}_2 = 3.36$, $\mathrm{err}_1 = 0.26$). Bottom left: Algorithm (D) (PSNR=36.79, $\mathrm{err}_2 = 3.68$, $\mathrm{err}_1 = 0.29$). Bottom right: Algorithm (E) (PSNR=38.58, $\mathrm{err}_2 = 2.99$, $\mathrm{err}_1 = 0.23$). All algorithms reduce the artifacts at the straight lines. However, the images at the top contain similar errors at the boundary of the circle. The images at the bottom have the best quality.

Figure 4.3: Original image of the second example and its degraded version.

Figure 4.4: Interpolation of the image in Fig. 4.3. Top left: cubic interpolation by the MATLAB procedure "griddata" (PSNR=30.18, $\mathrm{err}_2 = 7.89$, $\mathrm{err}_1 = 1.51$). Top right: Algorithm (A), (PSNR=31.61, $\mathrm{err}_2 = 6.69$, $\mathrm{err}_1 = 1.36$). Bottom left: Algorithm (B), (PSNR=34.08, $\mathrm{err}_2 = 5.03$, $\mathrm{err}_1 = 0.93$). Bottom right: Algorithm (C), (PSNR=33.98, $\mathrm{err}_2 = 5.09$, $\mathrm{err}_1 = 0.97$). The images at the top contain artifacts, in particular at the left window side. The algorithms at the bottom show a better performance and do not introduce these artifacts.

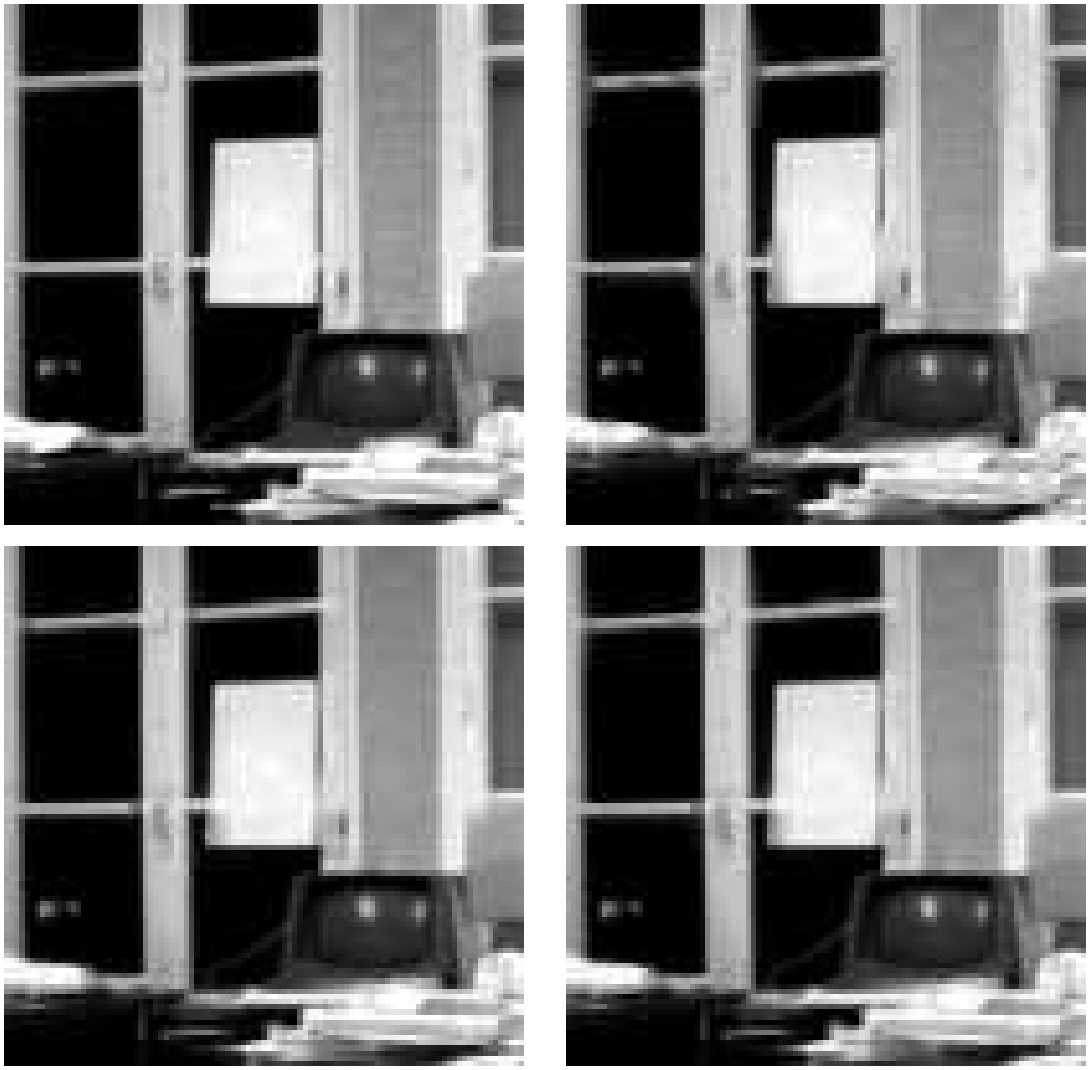


Figure 4.5: Details of the interpolated images of Fig. 4.4. Top left: original image. Top right: Algorithm (A), (PSNR=31.61, $\mathrm{err}_2 = 6.69$, $\mathrm{err}_1 = 1.36$). Bottom left: Algorithm (B), (PSNR=34.08, $\mathrm{err}_2 = 5.03$, $\mathrm{err}_1 = 0.93$). Bottom right: Algorithm (C), (PSNR=33.98, $\mathrm{err}_2 = 5.09$, $\mathrm{err}_1 = 0.97$). In contrast to the top right image, the images at the bottom have less errors at the horizontal and vertical window lines.

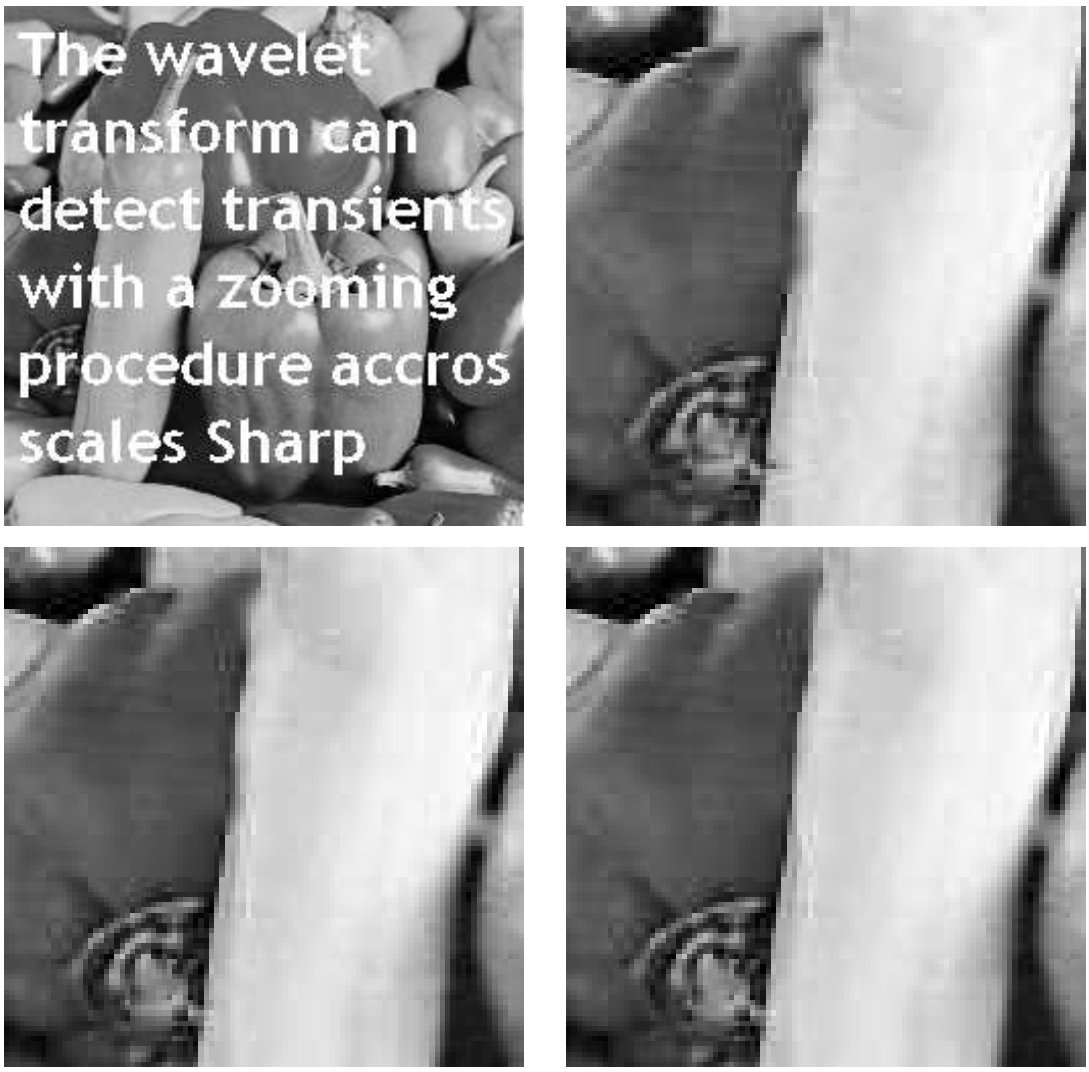


Figure 4.6: Interpolation results for the "peppers" image. Top left: degraded image. Top right: cubic interpolation. Bottom left: Algorithm (D). Bottom right: Algorithm (E). Algorithms (D) and (E) improve the quality at long edges.

CHAPTER 5

Application to the denoising of matrix fields

5.1 Introduction

In Chapter 3, we have studied the denoising of gray value images, i.e., scalar-valued functions. We have seen the good performance of the ROF model in edge-preserving restoration and the improvements achieved by using the related infimal convolution approach. Clearly, the question arises how we can transfer these methods to vector-valued and matrix-valued images. The classical example of vector-valued images are simply color images. We will not study them here, see, e.g., [66, Section 4.8] and the references therein, but proceed directly to *matrix-valued images*.

Matrix-valued data have gained significant importance in recent years. Some of their application are the following:

- First, *diffusion tensor magnetic resonance imaging (DT-MRI)* [14] is a modern but commonly used medical imaging technique that measures a 3×3 (or larger) positive semidefinite matrix-field: A so-called *diffusion tensor* is assigned to each voxel. This diffusion tensor describes the diffusive property of water molecules. Since water diffuses preferably along ordered tissue such as nerve fibers this matrix gives valuable information about the geometry and organization of the tissue under examination. Hence, the resulting matrix field plays a very important role in the diagnosis of multiple sclerosis and strokes. For detailed information about the acquisition of this type of data, the reader is referred to [3, 13] and the literature cited therein.

- Second, in many fields of technical science such as civil engineering, solid mechanics and geology, anisotropic behavior is often described satisfactorily by inertia, diffusion, stress, and permittivity tensors.

- Third, matrices/tensors have been recognized as a useful concept in image analysis itself [123]: The *structure tensor* [107], for instance, (also called Förstner interest operator, or scatter matrix) has been employed not only for corner detection [127], but also for texture analysis [180] and motion estimation [28]. Tensor voting, an

interesting recent tool for segmentation and grouping, also makes use of the tensor concept, see, e.g., [147, 156].

This variety of applications make it worthwhile to develop appropriate tools for the restoration and processing of tensor-valued, respectively matrix-valued data. Thus, we will transfer the successful techniques presented in Chapter 3 such as the ROF and the infimal convolution model to the matrix-valued setting. Clearly, the first approach would be to simply denoise the individual channels of the matrix-valued image independently. This is not a good strategy, however, since an object in a matrix-valued image will generally show characteristic features in more than one component. So, the main idea is that our functionals couple the different matrix channels. Moreover, unlike vectors, matrices can be multiplied providing matrix-valued polynomials and also functions of matrices. These useful notions rely decisively on the strong interplay between the different matrix entries and will give rise to a new *operator-based* regularization term. This is motivated by filtering methods for matrix fields based on matrix-valued nonlinear partial differential equations (PDEs) proposed in [44] for singular and in [45] for Perona-Malik-type diffusivity functions. Our method is the first variational approach for denoising tensor-valued data that takes the operator structure of matrices, in particular the operation of matrix multiplication, into account. Furthermore, they are also applicable to indefinite matrix fields.

Approaches to positive definite matrix field filtering with a differential geometric background have been suggested in [68, 124, 173, 210, 232]. In their setting the set of positive definite matrices is endowed with a structure of a manifold, and the methodology is geared towards application to DT-MRI data. For other smoothing techniques for DT-MRI data we refer, e.g., to [218, 228]. Comprehensive survey articles on the analysis of matrix fields utilizing a wide range of different techniques can be found in [222] and the literature cited therein.

This chapter, which is based on [194, 202, 195], is organized as follows: After giving the necessary preliminaries on matrix-valued data in Section 5.2, we consider *component-based* regularization terms related to the ROF model and its improvement via the infimal convolution in Section 5.3. These functionals couple the different matrix channels as originally proposed by [210]. In Section 5.4, we introduce a new *operator-based* functional and derive the corresponding Euler-Lagrange equation which contains the Jordan product of matrices. In contrast to the ordinary matrix product, the Jordan product of two symmetric matrices is again a symmetric matrix. Finally, in Section 5.5, we present numerical examples comparing the component-based and the operator-based approach as well as first-order and infimal convolution methods.

It should be remarked here that we do not study the noise statistics of a specific application here but assume white Gaussian noise. Note that our models perform well on real-world DT-MRI data, cf., Section 5.5. For noise models in the case of DT-MRI, we refer to [3, 15].

5.2 Preliminaries

Let $\mathrm{Sym}_m(\mathbb{R})$ be the vector space of symmetric $m \times m$ matrices. As before, we often reorder a matrix $A \in \mathbb{R}^{m,m}$ columnwise into a vector of length m^2 which we will denote in this chapter by $\mathrm{vec}A$. The space $\mathrm{Sym}_m(\mathbb{R})$ can be treated as a Euclidean vector space with respect to the trace inner product

$$\langle A, B\rangle := \operatorname{tr} AB = \langle \mathrm{vec}A, \mathrm{vec}B\rangle, \tag{5.1}$$

where $\langle\cdot,\cdot\rangle$ on the right-hand side denotes the Euclidean inner product in $\mathbb{R}^{m^2}$. Then,

$$\langle A, A\rangle = \operatorname{tr} A^2 = \|A\|_F^2 = \|\mathrm{vec}A\|_2^2$$

is the squared *Frobenius norm* of A. In $\mathrm{Sym}_m(\mathbb{R})$, the positive semi-definite matrices $\mathrm{Sym}_m^+(\mathbb{R})$ form a closed, convex set, whose interior consists of the positive definite matrices. More precisely, $\mathrm{Sym}_m^+(\mathbb{R})$ is a cone with base $\mathcal{B}$, see [11, 42, 43], i.e.,

$$\mathrm{Sym}_m^+(\mathbb{R}) = \mathbb{R}_{\geq 0}\,\mathcal{B}$$

and

$$\mathcal{B} := \{B \in \mathrm{Sym}_m^+(\mathbb{R}) : \operatorname{tr} B = 1\}.$$

Since $\mathcal{B}$ is a convex compact set in a finite-dimensional space it is, by the Krein-Milman theorem, the convex hull of its extreme points which are given by the rank 1 matrices vv^{T} with $\|v\|_2 = 1$. Thus, we have

$$\mathcal{B} = \mathrm{conv}\{vv^{\mathrm{T}} : v \in S^{n-1}\}, \tag{5.2}$$

where conv means that we consider the convex hull of the set on the right-hand side of (5.2). For $m = 2$, the above relations can be illustrated as follows: we embed $\mathrm{Sym}_2(\mathbb{R})$ into $\mathbb{R}^3$ by

$$A \mapsto a := \frac{1}{\sqrt{2}}\big(2A(1,2), A(1,1) - A(2,2), A(1,1) + A(2,2)\big)^{\mathrm{T}}. \tag{5.3}$$

This mapping is an isometry from $\mathrm{Sym}_2(\mathbb{R})$ equipped with the Frobenius norm onto $\mathbb{R}^3$ endowed with the Euclidean norm. For $A \in \mathrm{Sym}_2^+(\mathbb{R})$ with eigenvalues $\lambda_1, \lambda_2 \geq 0$, we have that

$$\begin{aligned} \lambda_1 + \lambda_2 &= \operatorname{tr}A = A(1,1) + A(2,2) = \sqrt{2}\,a_3 \geq 0, \\ \lambda_1\lambda_2 &= \det A = \frac{1}{4}\big((A(1,1) + A(2,2))^2 - (A(1,1) - A(2,2))^2 - 4A(1,2)^2\big) \geq 0. \end{aligned}$$

Hence, $A \in \mathrm{Sym}_2^+(\mathbb{R})$ if and only if $a_3 \geq 0$ and $\|(a_1, a_2)^{\mathrm{T}}\|_2 \leq a_3$, i.e., the symmetric positive semi-definite matrices form the cone $\mathcal{C}^3 := \{a \in \mathbb{R}^3 : \|(a_1, a_2)^{\mathrm{T}}\|_2 \leq a_3\}$ depicted in Fig. 5.1. Its base $\mathcal{B}$ is just the closed disc at the height $1/\sqrt{2}$ and the extreme points

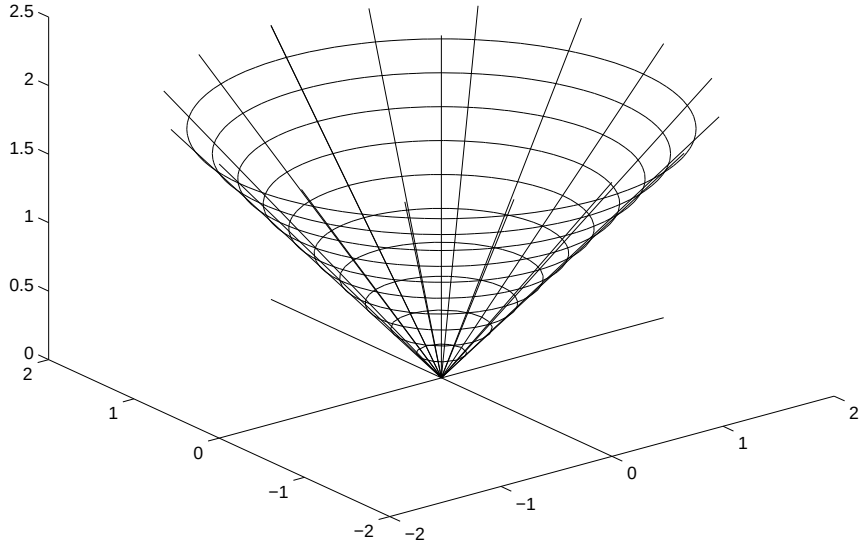


Figure 5.1: Cone of symmetric, positive semi-definite matrices via (5.3).

form the boundary of this disc. For our numerical examples we will further use that the positive definite matrices $A \in \mathrm{Sym}_2^+(\mathbb{R})$ can be visualized as ellipses

$$\{x \in \mathbb{R}^2 : x^{\mathrm{T}} A^{-2} x = 1\}$$

whose axes have just the length of the eigenvalues of A. Similarly, we can use ellipsoids to visualize the positive definite matrices in the case $m = 3$.

Matrices are can also be interpreted as (realizations of) linear operators and carry the corresponding features. In particular, they can be applied successively. Unfortunately, the original matrix multiplication does not preserve the symmetry of the matrices. The *Jordan-product* of matrices $A, B \in \mathrm{Sym}_m(\mathbb{R})$ defined by

$$A \bullet B = \frac{1}{2}(AB + BA) \tag{5.4}$$

preserves the symmetry of the matrices but not the positive semi-definiteness.

In the subsequent sections, we want to transfer the denoising methods from Chapter 3, in particular the ROF model, to matrix-valued images. While it seem straightforward to replace the squares in the data-fitting of the scalar-valued ROF functional (3.11) by the squared Frobenius norm, the regularization term may be chosen in different ways. In the following, let $F : \mathbb{R}^2 \supset \Omega \to \mathrm{Sym}_m(\mathbb{R})$ be a matrix field.

5.3 Component-based regularization

In this section, we transfer the ROF model (3.11) to matrix-valued images in a way that emphasizes the individual matrix components. Assume that for each channel the scalar-valued function $F_{i,j}$, $i,j = 1,\dots,m$, corresponding to F lies in the Sobolev space $W^1_{1,1}(\Omega)$. Instead of (3.11) we can then deal with

$$\operatorname*{argmin}_U \int_\Omega \frac{1}{2}\|U-F\|_F^2 + \lambda \underbrace{\varphi\left(\operatorname{tr}\left(U_x^2+U_y^2\right)\right)}_{=:J(U)} dxdy, \tag{5.5}$$

where the partial derivatives are taken componentwise and φ is, e.g., given by $\varphi(s^2)=\sqrt{s^2}$. The penalizing term $J(U)$ in (5.5) was first used by Deriche and Tschumperlé [210]. Note that λJ then plays the role of $\Phi \circ D$ in (3.4). Rewriting J as

$$J(U) = \int_\Omega \varphi\big(\|U_x\|_F^2 + \|U_y\|_F^2\big)\, dxdy = \int_\Omega \varphi\big(\sum_{j,k=1}^{n} \nabla u_{jk}^{\mathrm{T}} \nabla u_{jk}\big)\, dxdy, \tag{5.6}$$

we see its component-based structure implied by the Frobenius norm. However, due to the sum on the right–hand side, φ is applied to the *coupled* matrix coefficients. By [38], the Euler–Lagrange equation of (5.6) is given by

$$0 = U - F + \lambda\left(\partial_x(\varphi'(\operatorname{tr}(U_x^2+U_y^2))U_x + \partial_y(\varphi'(\operatorname{tr}(U_x^2+U_y^2))U_y\right). \tag{5.7}$$

For the ROF functional we can choose the approximation $\varphi(s^2)=\sqrt{s^2+\varepsilon^2}$ for a small parameter ε.

For computations we consider the discrete counterpart of (5.5) and we use here the forward difference discretization (3.18) of the gradient. We thus minimize the following problem

$$\operatorname*{argmin}_U \sum_{i,j=0}^{n-1} \frac{1}{2}\,\|U(i,j)-F(i,j)\|_F^2 \;+\; \lambda\, J(U), \tag{5.8}$$

$$J(U) \;:=\; \sum_{i,j=0}^{n-1} \big(\|U(i,j)-U(i-1,j)\|_F^2 + \|U(i,j)-U(i,j-1)\|_F^2\big)^{1/2}$$

over the set of all discrete matrix fields $U : \mathbb{Z}_n^2 \to \mathrm{Sym}_m^+(\mathbb{R})$ where $\mathbb{Z}_n^2 = \{1,\dots,n\}\times\{1,\dots,n\}$. We again assume mirrored boundary condition in (5.8), i.e., we set $U(-1,j) = U(0,j)$ and $U(i,-1)=U(i,0)$ for all $i,j=0,\dots,n-1$. The functional in (5.8) is strictly convex and thus has a unique minimizer.

We say that the discrete matrix field $F : \mathbb{Z}_n^2 \to \mathrm{Sym}_m^+(\mathbb{R})$ has all eigenvalues in an interval $\mathcal{I}$ if all the eigenvalues of every matrix $F(i,j)$ of the field lie in $\mathcal{I}$. By the following proposition, the minimizer of (5.8) preserves positive definiteness.

Proposition 5.3.1 *Let all eigenvalues of $F : \mathbb{Z}_n^2 \to \mathrm{Sym}_m^+(\mathbb{R})$ be contained in the interval $[\lambda_{\min}, \lambda_{\max}]$. Then the minimizer $\hat{U}$ of (5.8) has all eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$.*

Proof: Using that the minimal and maximal eigenvalues $\lambda_{\min}(A), \lambda_{\max}(A)$ of a symmetric matrix A fulfill

$$\lambda_{\min}(A) = \min_{\|v\|=1} v^{\mathrm{T}} A\, v, \quad \lambda_{\max}(A) = \max_{\|v\|=1} v^{\mathrm{T}} A\, v,$$

it is easy to check that the set $\mathcal{C}$ of matrices having all eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$ is convex and closed. Let $\mathcal{J}$ be the whole functional in (5.8). Assume that some matrices $\hat{U}(i,j)$ are not contained in $\mathcal{C}$. Let $P\hat{U}(i,j)$ denote the orthogonal projection (w.r.t. the Frobenius norm) of $\hat{U}(i,j)$ onto $\mathcal{C}$. Then we obtain by the projection theorem [72, p. 269] that

$$\begin{aligned} \|F(i,j) - P\hat{U}(i,j)\|_F &\leq \|F(i,j) - \hat{U}(i,j)\|_F, \\ \|P\hat{U}(i,j) - P\hat{U}(k,l)\|_F &\leq \|\hat{U}(i,j) - \hat{U}(k,l)\|_F. \end{aligned}$$

Consequently, $\mathcal{J}(P\hat{U}) \leq \mathcal{J}(\hat{U})$ which contradicts our assumption since the minimizer is unique. This completes the proof. □

To better see the connection to scalar-valued denoising we rewrite (5.8) in matrix-vector form. To this end, let $N = n^2$ and $M := m(m+1)/2$. We reshape $F : \mathbb{Z}_n^2 \to \mathrm{Sym}_m(\mathbb{R})$ into the vector

$$f := \begin{pmatrix} \varepsilon_{1,1} & \mathrm{vec}\,(F_{1,1}) \\ \vdots & \\ \varepsilon_{1,m} & \mathrm{vec}\,(F_{1,m}) \\ \varepsilon_{2,2} & \mathrm{vec}\,(F_{2,2}) \\ \vdots & \\ \varepsilon_{2,m} & \mathrm{vec}\,(F_{2,m}) \\ \vdots & \\ \varepsilon_{m,m} & \mathrm{vec}\,(F_{m,m}) \end{pmatrix} \in \mathbb{R}^{MN},$$

where $F_{k,l} := (F_{k,l}(i,j))_{i,j=0}^{n-1}$ and $\varepsilon_{k,l} := \begin{cases} \sqrt{2} & \text{for } k \neq l \\ 1 & \text{otherwise} \end{cases}$.

Then (5.8) becomes

$$\operatorname*{argmin}_{u \in \mathbb{R}^{MN}} \{ \frac{1}{2} \|f - u\|_2^2 \; + \; \lambda \| |(I_M \otimes D)\, u| \|_1 \}, \tag{5.9}$$

where $D := D_2$ is the forward difference discretization of the gradient defined in (3.18). We immediately see that problem (5.9) has just the structure of (3.12) with the larger matrix $I_M \otimes D \in \mathbb{R}^{2MN, MN}$. Thus we can apply all the methods presented in Chapter 3 to solve (5.9) or its dual problem.

Similarly, we can transfer the infimal convolution approach to the matrix-valued setting and use the algorithms described in Section 3.6. Obviously, we have to find

$$\operatorname*{argmin}_{u_1, u_2 \in \mathbb{R}^{MN}} \frac{1}{2} \|u_1 + u_2 - f\|_2^2 + \lambda_1 \| |(I_M \otimes D)\, u_1| \|_1 + \lambda_2 \| |(I_M \otimes \mathcal{D}_2)\, u_1| \|_1,$$

where D and $\mathcal{D}_2$ are defined as in Section 3.6. In the same way, we can rewrite the modified dual problem (3.62) as follows

$$\|f - (I_M \otimes \mathcal{D}_2^{\mathrm{T}})\, V\|_2^2 \;\to \min \quad \text{s.t.} \qquad \| \,|\left(I_M \otimes \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}\right) b|\, \|_\infty \le \lambda_1, \quad \| \,|b|\, \|_\infty \le \lambda_2. \tag{5.10}$$

5.4 Operator-based regularization

In this section, we introduce a regularization term that emphasizes the operator structure of matrices. For $A \in \mathrm{Sym}_m(\mathbb{R})$ with eigenvalue decomposition $A = Q\Sigma Q^{\mathrm{T}}$, let $\varphi(A) = Q\varphi(\Sigma)Q^{\mathrm{T}}$, where $\Sigma := \mathrm{diag}\,(\sigma_1, \ldots, \sigma_n)$ and $\varphi(\Sigma) := \mathrm{diag}\,(\varphi(\sigma_1), \ldots, \varphi(\sigma_n))$. We consider the following minimization problem

$$\operatorname*{argmin}_U \int_\Omega \frac{1}{2}\|U - F\|_F^2 + \lambda\, \mathrm{tr}\left(\varphi(U_x^2 + U_y^2)\right)\, dxdy. \tag{5.11}$$

In contrast to (5.5), the trace is taken *after* applying φ to the matrix $U_x^2 + U_y^2$.

We want to show now that the functional in (5.11) with $\varphi := \sqrt{\cdot}$ is strictly convex. Since φ operates on the eigenvalues of the corresponding matrix it is useful to recall the following classical result which shows the correspondence between matrix norms and functions on the singular values: A matrix norm $\|\cdot\|$ on the space of complex $r \times s$ matrices is called *unitarily invariant* if $\|UAV\| = \|A\|$, for all $A \in \mathbb{C}^{r,s}$ and all unitary matrices $U \in \mathbb{C}^{r,r}$ and $V \in \mathbb{C}^{s,s}$. Furthermore, we need the notion of a *symmetric gauge function* on $\mathbb{C}^t$. It is defined as a vector norm on $\mathbb{C}^r$ with the following two additional properties: First, it is invariant to permutation of components of the input vector and, second, it is unchanged when we take the absolute value of the components of the input vector.

We have the following theorem which can be traced back to [216]. Note that there exist many related results, e.g., for symmetric matrices [85]. For an overview of this topic, we refer to [143].

Theorem 5.4.1 *Consider the space of complex $r \times s$ matrices. Let $t := \min\{r, s\}$ and assume that the function f on $\mathbb{C}^t$ is a symmetric gauge function. For any $A \in \mathbb{C}^{r,s}$, define $\|A\| = f(\sigma)$, where $\sigma = (\sigma_1, \ldots, \sigma_t)$ are the singular values of A. Then, $\|\cdot\|$ is a unitarily invariant matrix norm. The converse is also true.*

Proof: Compare, e.g., [130, p. 439].

A classical example of a matrix norm defined by Theorem 5.4.1 is the *trace norm* $\|\cdot\|_{\mathrm{tr}}$, also called the *nuclear norm*. It is defined as the sum of the singular values of a matrix or, equivalently, as

$$\|A\|_{\mathrm{tr}} = \mathrm{tr}(\sqrt{A^*A}), \quad \text{for} \quad A \in \mathbb{C}^{s,t}.$$

Remark 5.4.2 *Recently, there is a lot of interest in matrix completion problems for which the trace norm and its subdifferential is important, cf.,* [48, 209]*. For details about the subdifferential of unitarily invariant matrix norms, see also* [219]*.*

Proposition 5.4.3 *For given* $F : \mathbb{R}^2 \supset \Omega \to \mathrm{Sym}_m(\mathbb{R})$ *and* $\varphi(s^2) = \sqrt{s^2}$*, the functional in* (5.11) *is strictly convex.*

Proof: Since $\|U - F\|_F^2$ is strictly convex, it remains to show that the functional

$$J(U) := \mathrm{tr}\left(\sqrt{U_x^2 + U_y^2}\right)$$

is convex. Moreover, since J is positively homogeneous we only have to prove that J is subadditive, cf. [30, p. 34], i.e.,

$$J(\tilde{U} + U) \leq J(\tilde{U}) + J(U).$$

This can be written as

$$\mathrm{tr}\left(\sqrt{(\tilde{U}_x + U_x)^2 + (\tilde{U}_y + U_y)^2}\right) \leq \mathrm{tr}\left(\sqrt{\tilde{U}_x^2 + \tilde{U}_y^2}\right) + \mathrm{tr}\left(\sqrt{U_x^2 + U_y^2}\right).$$

By definition of the trace norm, we have for the symmetric matrices $\tilde{U}_x, \tilde{U}_y, U_x, U_y$ that

$$\| \begin{pmatrix} \tilde{U}_x + U_x \\ \tilde{U}_y + U_y \end{pmatrix} \|_{\mathrm{tr}} = \mathrm{tr}\left(\sqrt{(\tilde{U}_x + U_x)^2 + (\tilde{U}_y + U_y)^2}\right)$$

Since $\|\cdot\|_{\mathrm{tr}}$ is a norm it follows that

$$\begin{aligned} \| \begin{pmatrix} \tilde{U}_x + U_x \\ \tilde{U}_y + U_y \end{pmatrix} \|_{\mathrm{tr}} &\leq \| \begin{pmatrix} \tilde{U}_x \\ \tilde{U}_y \end{pmatrix} \|_{\mathrm{tr}} + \| \begin{pmatrix} U_x \\ U_y \end{pmatrix} \|_{\mathrm{tr}} \\ &= \mathrm{tr}(\sqrt{\tilde{U}_x^2 + \tilde{U}_y^2}) + \mathrm{tr}(\sqrt{U_x^2 + U_y^2}) \end{aligned}$$

and we are done. □

Remark 5.4.4 *The solution of* (5.11) *does in general not preserve positive definiteness. For an example, see* [202]*.*

Let us assume now that φ is differentiable, e.g., $\varphi = \sqrt{\cdot + \varepsilon^2}$. The next proposition shows that the functional (5.11) has an interesting Gâteaux derivative.

Proposition 5.4.5 *Let* φ *be a differentiable function. Then, the Euler-Lagrange equations for minimizing the functional in* (5.11) *are given by*

$$0 = U - F + \lambda \left(\partial_x \left(\varphi'(U_x^2 + U_y^2) \bullet U_x\right) + \partial_y \left(\varphi'(U_x^2 + U_y^2) \bullet U_y\right)\right), \tag{5.12}$$

where $\bullet$ *is the Jordan product defined in* (5.4)*.*

Proof: Let $h(U_x, U_y) := \operatorname{tr}\left(\varphi(U_x^2 + U_y^2)\right)$. The Euler-Lagrange equations of (5.11) are given, for $i, j = 1, ..., n$ and $i \geq j$, by

$$0 = \frac{\partial}{\partial u_{ij}} \|U - F\|_F^2 \; - \; \lambda \left(\frac{\partial}{\partial x} \left(\frac{\partial h}{\partial u_{ijx}} \right) + \frac{\partial}{\partial y} \left(\frac{\partial h}{\partial u_{ijy}} \right) \right).$$

For a scalar-valued function f and an $n \times n$ matrix X, we set $\frac{\partial f(X)}{\partial X} := \left(\frac{\partial f(X)}{\partial x_{ij}} \right)_{i,j=1}^{n}$. Then, by symmetry of F and U, the Euler-Lagrange equations can be rewritten in matrix-vector form as

$$W_n \circ \frac{U - F}{\lambda} = \frac{1}{2} \left(\frac{\partial}{\partial x} \left(\frac{\partial h}{\partial U_x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial h}{\partial U_y} \right) \right), \tag{5.13}$$

where W_n denotes the $n \times n$ matrix with diagonal entries 1 and other coefficients 2, and $A \circ B$ stands for the *Hadamard product* (componentwise product) of A and B.
We consider $f(X) := \operatorname{tr} \varphi(X^2)$. Then we obtain by [148, p. 178] and $\operatorname{tr}(A^{\mathrm{T}}B) = (\mathrm{vec}A)^{\mathrm{T}}\mathrm{vec}B$ that

$$\begin{aligned} \mathrm{vec}\frac{\partial f(X)}{\partial X} &= \mathrm{vec} \left(\operatorname{tr} \left(\varphi'(X^2) \, \frac{\partial (X^2)}{\partial x_{ij}} \right) \right)_{i,j=1}^{n} \\ &= \mathrm{vec} \left((\mathrm{vec}\psi)^{\mathrm{T}} \mathrm{vec} \frac{\partial (X^2)}{\partial x_{ij}} \right)_{i,j=1}^{n} \end{aligned}$$

where $\psi := \varphi'(X^2)$. By [148, p. 182] and since ψ is symmetric, this can be rewritten as

$$\mathrm{vec}\frac{\partial f(X)}{\partial X} = \mathrm{vec}\; W_n \circ ((I_n \otimes X) + (X \otimes I_n)) \,\mathrm{vec}\; \psi.$$

Using that $\mathrm{vec}(ABC) = (C^{\mathrm{T}} \otimes A)\mathrm{vec}B$ we infer that

$$\mathrm{vec}\frac{\partial f(X)}{\partial X} = \mathrm{vec}W_n \circ \mathrm{vec}(X\psi + \psi X).$$

This implies that

$$\frac{\partial f(X)}{\partial X} = 2\, W_n \circ (\psi \bullet X). \tag{5.14}$$

Applying (5.14) with $f(U_x) := h(U_x, U_y)$ and $f(U_y) := h(U_x, U_y)$, respectively, in (5.13) we obtain the assertion. □

We apply Proposition 5.4.5 to compute a minimizer of (5.11) by using a difference method to solve the corresponding reaction–diffusion equation for $t \to \infty$

$$U_t = U - F + \lambda \left(\partial_x \left(\varphi'(U_x^2 + U_y^2) \bullet U_x \right) + \partial_y \left(\varphi'(U_x^2 + U_y^2) \bullet U_y \right) \right) \tag{5.15}$$

with φ defined as $\varphi(s^2) = \sqrt{s^2 = \varepsilon^2}$, homogeneous Neumann (mirrored) boundary conditions and initial value F. More precisely, we use the iterative scheme

$$U^{(k+1)} = (1-\tau)U^{(k)} + \tau F + \tau\lambda\left(\partial_x\left(G^{(k)} \bullet U_x^{(k)}\right) + \partial_y\left(G^{(k)} \bullet U_y^{(k)}\right)\right) \tag{5.16}$$

with sufficiently small time step size τ and $G^{(k)} := \varphi'((U_x^{(k)})^2 + (U_y^{(k)})^2)$. The inner derivatives including those in G are approximated by forward differences and the outer derivatives by backward differences so that the penalizing term becomes

$$\begin{aligned} &\frac{1}{h_1}\left(G(i,j) \bullet \frac{U(i+1,j) - U(i,j)}{h_1} - G(i-1,j) \bullet \frac{U(i,j) - U(i-1,j)}{h_1}\right) \\ + \quad &\frac{1}{h_2}\left(G(i,j) \bullet \frac{U(i,j+1) - U(i,j)}{h_2} - G(i,j-1) \bullet \frac{U(i,j) - U(i,j-1)}{h_2}\right), \end{aligned}$$

where h_i, $i = 1, 2$ denote the pixel distances in x– and y–direction. Alternatively, we have also worked with symmetric differences for the derivatives. In this case we have to replace e.g. $G(i,j)$ in the first summand by $\tilde{G}(i+1,j) + \tilde{G}(i,j))/2$ and $\tilde{G}$ is now computed with symmetric differences.

Finally, we mention that a diffusion equation related to (5.15) was examined in [44]. Moreover, in [221] an anisotropic diffusion concept for matrix fields was presented where the function φ was also applied to a matrix.

5.5 Numerical results

Finally, we present numerical results demonstrating the performance of the different methods. The problems in Section 5.3 can again be solved via the iterative algorithms proposed in Chapter 3 since they have essentially the same structure. Another possibility is to find a solution via second-order cone programming (SOCP), see, e.g., [202] where we use the software package MOSEK.

SOCP [4, 118, 145] amounts to minimize a linear objective function subject to the constraints that several affine functions of the variables have to lie in a *second-order cone* $\mathcal{C}^{n+1} \subset \mathbb{R}^{n+1}$ defined by the convex set

$$\mathcal{C}^{n+1} = \left\{ \begin{pmatrix} x \\ \bar{x}_{n+1} \end{pmatrix} = (x_1, \ldots, x_n, \bar{x}_{n+1})^{\mathrm{T}} : \|x\|_2 \leq \bar{x}_{n+1} \right\}.$$

With this notation, the general form of a SOCP is given by

$$\inf_{x \in \mathbb{R}^n} \; f^{\mathrm{T}} x \quad \text{s.t.} \; \begin{pmatrix} A_i x + b_i \\ c_i^{\mathrm{T}} x + d_i \end{pmatrix} \in \mathcal{C}^{n+1}, \; i = 1, \ldots, r. \tag{5.17}$$

Alternatively, one can also use the rotated version of the standard cone:

$$\mathcal{K}^{n+2} := \left\{ \left(x, \bar{x}_{n+1}, \bar{x}_{n+2}\right)^{\mathrm{T}} \in \mathbb{R}^{n+2} : \|x\|_2^2 \leq 2\, \bar{x}_{n+1} \bar{x}_{n+2} \right\}.$$

This allows us to incorporate quadratic constraints. Problem (5.17) is a convex program for which efficient, large scale solvers are available [159]. Note that the ROF problem discussed in Chapter 3 can also be solved via SOCP, cf., [118]. Second-order cone programs are generally solved by applying Newton steps to a primal-dual formulation of the given problem. The connection between SOCP for ROF and the special Newton method of [64] was recently analyzed in [234]. For rewriting our minimization problems as a SOCP, see [202].

We start by comparing the component-based regularization with the operator-based regularization. First we are interested in the 1D matrix–valued function $F : \mathbb{Z}_{16} \to \mathrm{Sym}_2^+(\mathbb{R})$ in Fig. 5.2. We added white Gaussian noise with standard deviation 0.1 to all components of the original data in $[0, 1]$. Then we computed the minimizer of the component-based functional (5.8) (left) and of the operator-based functional (5.11) (right). The latter was calculated using the fact that the operator-based functional can be rewritten for tensor-valued 1D signals as

$$\operatorname*{argmin}_U \int_\Omega \|U - F\|_F^2 + \lambda \operatorname{tr}|U_x|\, dx$$

with

$$\operatorname{tr}|U| = \max\{\left(4u_{12}^2 + (u_{11} - u_{22})^2\right)^{1/2}, |u_{11} + u_{22}|\}$$

for $U \in \mathrm{Sym}_2^+(\mathbb{R})$, cf. [202]. The middle of the figure shows the Frobenius norm of the difference between the original and the denoised signal $(\sum_{i=1}^N \|\hat{U}(i) - F(i)\|_F^2)^{1/2}$ in dependence on the regularization parameter λ. We remark that the shape of the curve and its minimal point do not change if we use the error measure $\sum_{i=1}^N \|\hat{U}(i) - F(i)\|_F$ instead. The actual minima with respect to the Frobenius norm are given by min = 0.2665 at $\lambda = 0.8$ for (5.8) and by min = 0.2276 at $\lambda = 0.8$ for (5.11). The denoised signals corresponding to the smallest error in the Frobenius–norm are shown at the bottom of Fig. 5.2. It appears that the operator-based method performs slightly better with respect to these error norms. The visual results confirm this impression. The larger ellipses obtained by the first method (5.8) slightly overlap while there are gaps between the smaller ones. We do not have this effect for the minimizer of (5.11) on the right-hand side.

Next we consider the 2D matrix–valued function $F : \mathbb{Z}_{32}^2 \to \mathrm{Sym}_2^+(\mathbb{R})$ in Fig. 5.3. To all components of the original data in [0,2] we added white Gaussian noise with standard deviation 0.6. As in the previous example, we compare the minimizer of the component-based approach (5.5), resp. (5.8), with those of the operator-based approach (5.11). To compute the minimizer of the latter, we used the time step size $\tau = 0.00025$ in the reaction–diffusion equation (5.16). The iterations were stopped when the relative error in the ℓ_2-norm between two consecutive iterations became smaller than 10^{-8} (approximately 20000 iterations) although the result becomes visually static much earlier. The middle row of the figure contains the error plots for both methods. The actual minima w.r.t. the Frobenius norm are given by min = 12.19 at $\lambda = 1.75$ for (5.8) and by min = 10.79 at $\lambda = 1.2$ for (5.11). Hence, with respect to the computed errors the operator-based method outperforms the component-based one. The corresponding denoised images are shown in the bottom row of the figure.

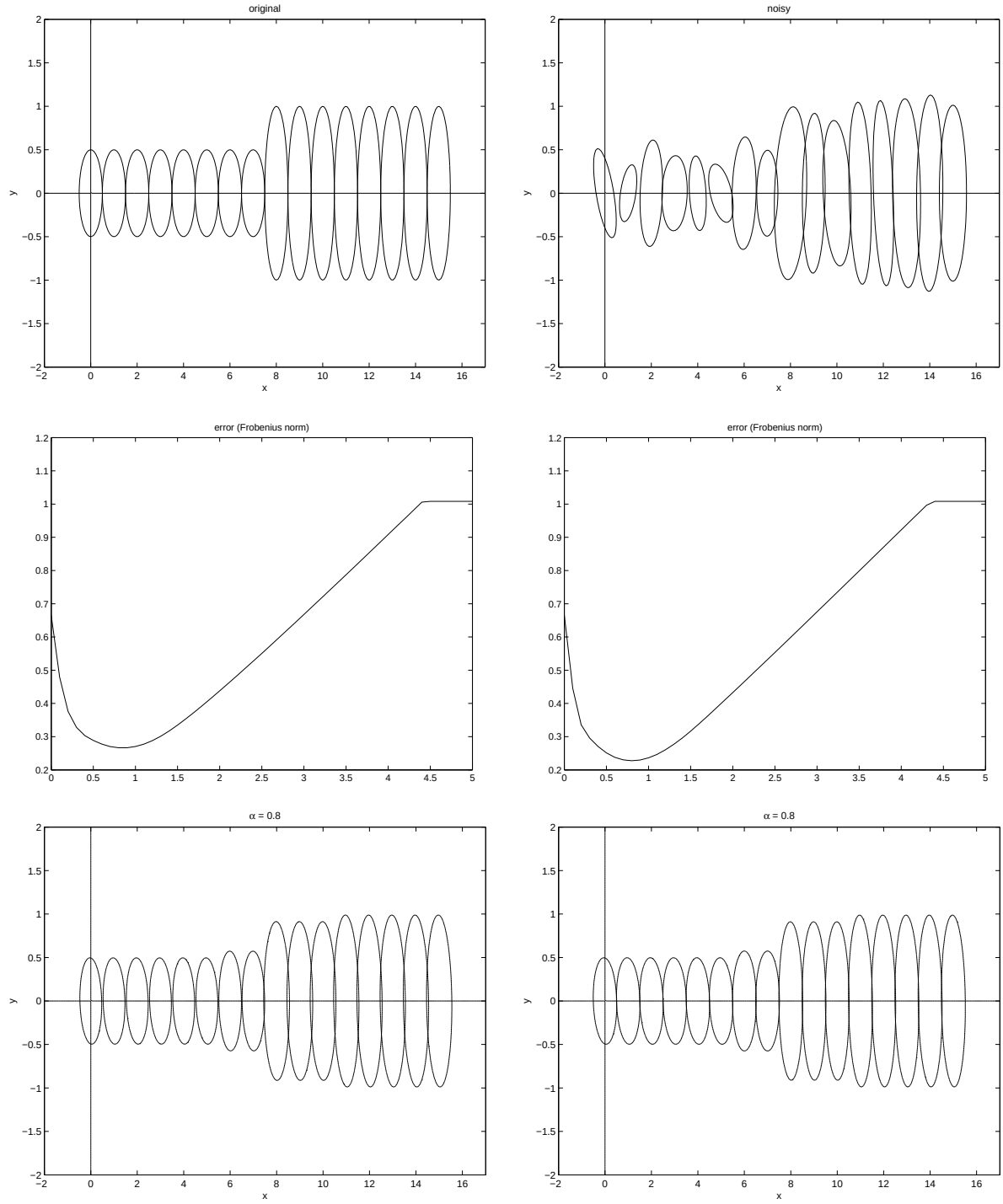


Figure 5.2: Denoising of a matrix–valued signal. **Top**: Original signal (left), noisy signal (right). **Middle**: Error of the Frobenius norm in dependence on the regularization parameter λ for the minimizers of the component-based functional (5.8) (left) and the operator-based functional (5.11) (right). **Bottom**: Denoised image for λ corresponding to the smallest error in the Frobenius norm for the component-based functional (left) and the operator-based functional (right).

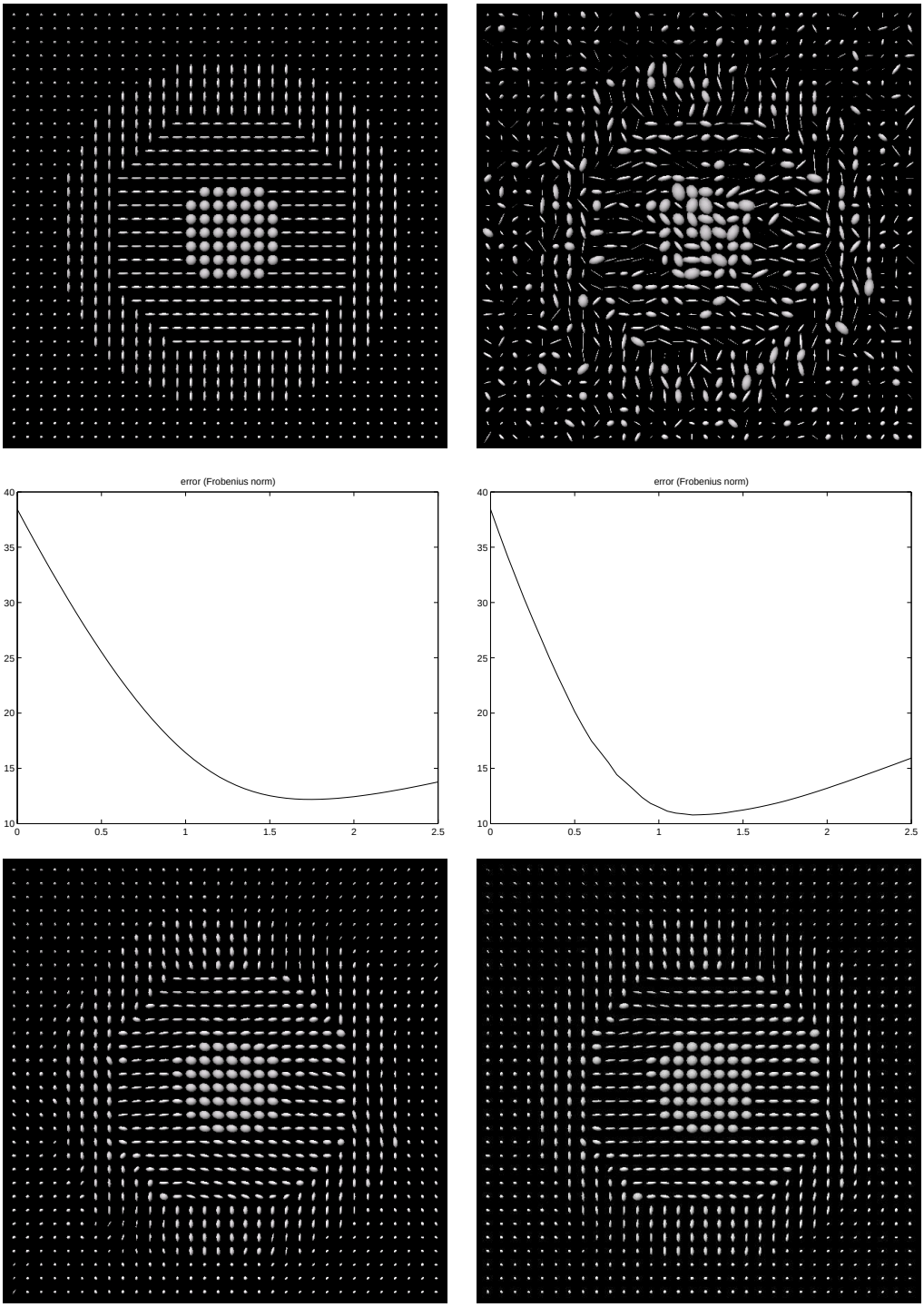


Figure 5.3: Denoising of a $\mathrm{Sym}_2(\mathbb{R})$–valued image. **Top**: Original image (left), noisy image (right). **Middle**: Error of the Frobenius norm in dependence on the regularization parameter λ for the minimizers of the component-based functional (5.8) (left) and the operator-based functional (5.11) (right). **Bottom**: Denoised image for λ corresponding to the smallest error in the Frobenius norm for the component-based functional (left) and the operator-based functional (right).

In the following two examples, we consider bivariate matrix-valued functions which map to $\mathrm{Sym}_3(\mathbb{R})$. We use ellipsoids to visualize this kind of data as described in Section 5.2. Furthermore, the color of the ellipsoid associated with a matrix A is chosen with respect to the normalized eigenvector corresponding to the largest eigenvalue of A. Fig. 5.4 (top) shows a function $F : \mathbb{Z}_{32}^2 \to \mathrm{Sym}_3(\mathbb{R})$. As before, we added white Gaussian noise to all components. The matrix components of the original data lie in the interval $[0, 1.2]$ and the standard deviation of the Gaussian noise is 0.23. The denoising results are displayed in the last two rows of Fig. 5.4. The smallest error for the component-based method (5.8), measured in the Frobenius-norm, is 7.037 and was obtained for the regularization parameter $\lambda = 0.60$. In addition, we considered the minimizer of the infimal convolution approach (5.10). The optimal regularization parameters were found to be $\lambda_1 = 0.54$ and $\lambda_2 = 1.51$ for this method. As in the scalar-valued case, the staircasing artifacts are now reduced and the corresponding Frobenius-norm error is only 6.087. So, we see that the infimal convolution approach is also suited for matrix-valued data.

In our final experiment, we applied the two component-based methods (5.8) and (5.10) to a larger data set. Fig. 5.5 shows the original data and the minimizers of (5.8) and (5.10). The components of the original data lie in $[-4000, 7000]$ and we used the regularization parameters $\lambda = 600$ for (5.8) and $\lambda_1 = 500$, $\lambda_2 = 600$ for (5.10), respectively.

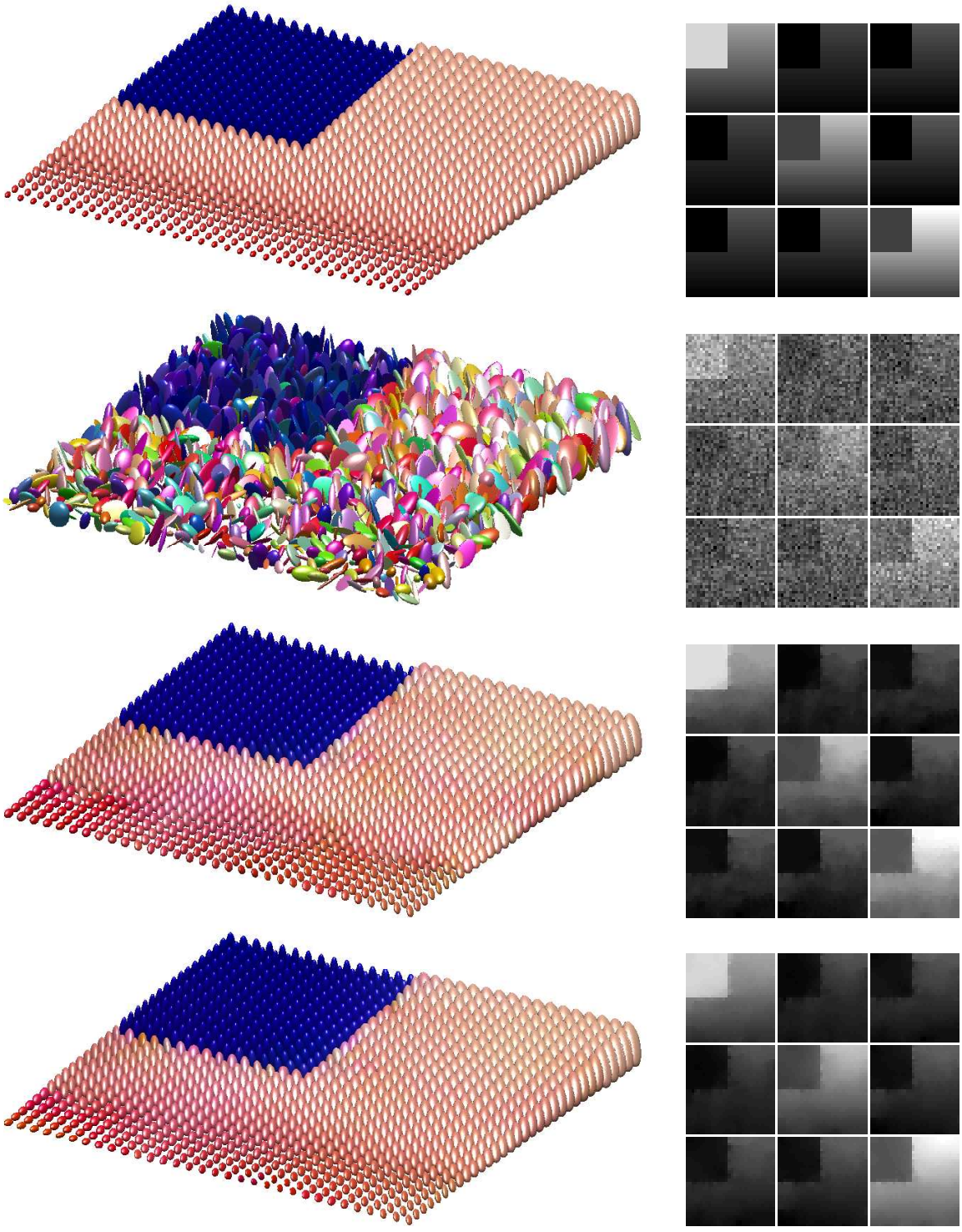


Figure 5.4: Denoising of a $\mathrm{Sym}_3(\mathbb{R})$-valued image. **Top to Bottom**: Original image, noisy image, minimizer of the component-based method (5.8) for $\lambda = 0.60$, minimizer of the component-based infimal convolution approach (5.10) with parameters $\lambda_1 = 0.54$, $\lambda_2 = 1.51$. Visualization: ellipsoids (left), components of the matrix-valued data (right).

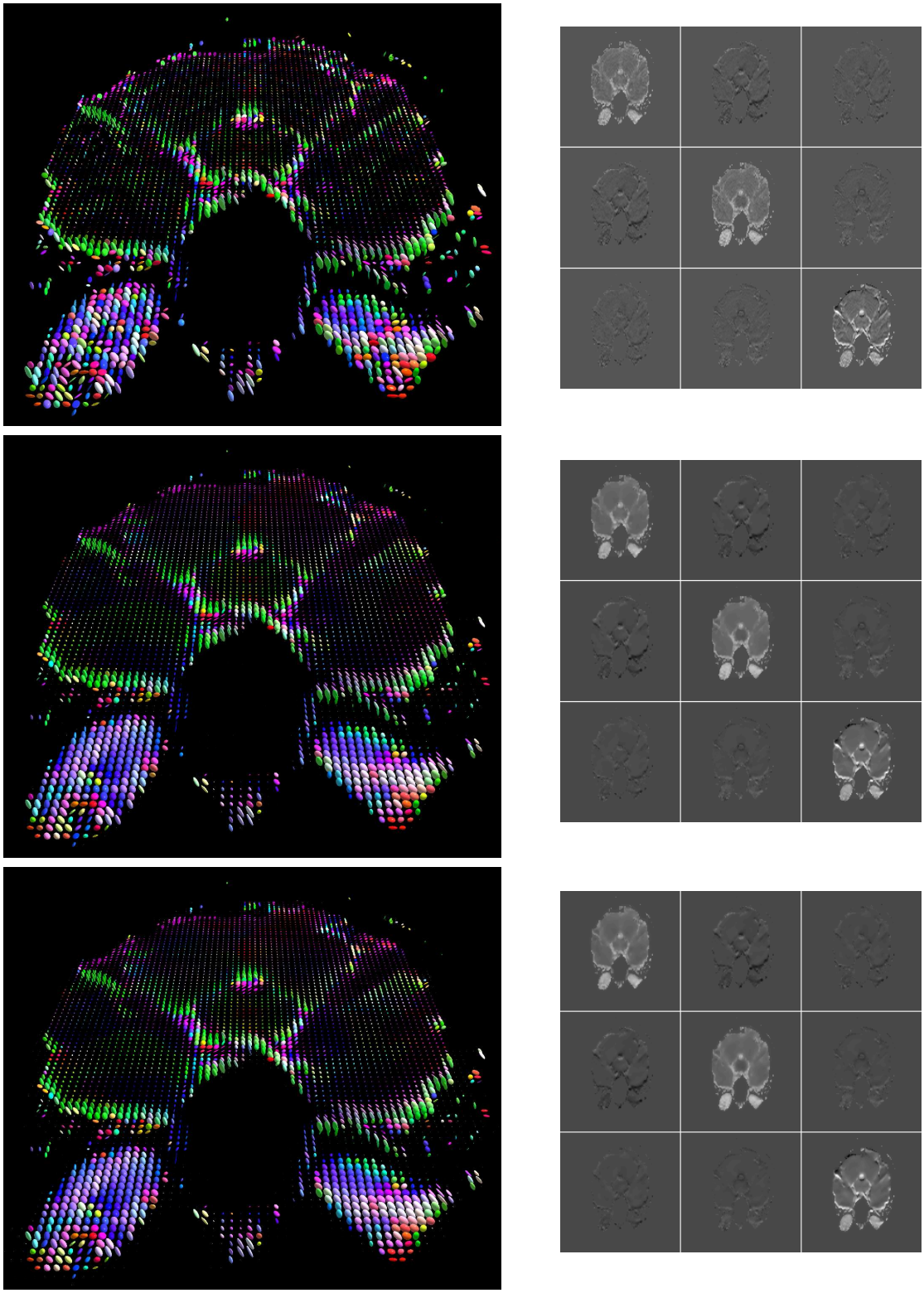


Figure 5.5: Denoising of a real-world DT-MRI matrix field with values in $\mathrm{Sym}_3(\mathbb{R})$. **Top**: Original image. **Middle**: Minimizer of the component-based method (5.8), $\lambda = 600$. **Bottom**: Minimizer of the infimal convolution approach (5.10) for $\lambda_1 = 500$, $\lambda_2 = 600$.

APPENDIX A

Wavelet-frames

At different places in this paper we consider representations of an image in terms of its frame coefficients and especially wavelet-frame coefficients. Disregarding the low-pass coefficients of such an expansion also gives rise to interesting discretizations of derivatives.

A.1 Frames

The notion of frames was introduced in [93]. For details and proofs of the results below we refer, e.g., to [71, 82]. Let us first work in a general Hilbert space setting. The sequence $(u_k)_{k\in\mathbb{N}}$ in a Hilbert space H is called a *frame* of H if there exist so-called frame bounds $a, b > 0$ with

$$a\|u\|^2 \leq \sum_{k=1}^{\infty} |\langle u, u_k\rangle|^2 \leq b\|u\|^2, \quad \forall u \in H. \tag{A.1}$$

A frame is called a *tight frame* if $a = b$ and a *Parseval frame* if we have the normalization $a = b = 1$. The *analysis operator* W of a frame returns the frame coefficients, i.e.,

$$W(u) = (\langle u, u_k\rangle)_{k\in\mathbb{N}}.$$

For a given sequence $(c_k)_{k\in\mathbb{N}}$ in $\ell_2(\mathbb{N})$ the adjoint operator W^* of W, called the *synthesis operator*, is defined as

$$W^*((c_k)_{k\in\mathbb{N}}) = \sum_{k=1}^{\infty} c_k u_k.$$

Hence, we can write (A.1) in the equivalent form

$$aI \leq W^*W \leq bI, \tag{A.2}$$

where W^*W is called the *frame operator*. Since the frame operator is continuously invertible we can define the canonical *dual frame* $(\tilde{u}_k)_{k\in\mathbb{N}} := ((W^*W)^{-1}u_k)_{k\in\mathbb{N}}$ corresponding to

$(u^{(k)})_{k\in\mathbb{N}}$ whose frame bounds are given by $\frac{1}{b}$ and $\frac{1}{a}$. The frame operator of the canonical dual frame is then given by $\widetilde{W} = W(W^*W)^{-1}$ and a short calculation shows that

$$\widetilde{W}^*W = W^*\widetilde{W} = I. \tag{A.3}$$

From (A.3) we deduce two important ways to make use of frames. First, we have for each element $u \in H$ that

$$u = W^*\widetilde{W}(u) = \sum_{i=1}^{\infty} \langle u, \tilde{u}_k\rangle u_k,$$

i.e., the frame spans the space H and the coefficients of this expansion can be given in terms of the dual frame. Second, it holds that

$$u = \widetilde{W}^*W(u) = \sum_{i=1}^{\infty} \langle u, u_k\rangle \tilde{u}_k,$$

which shows that we can reconstruct the element u for the given frame coefficients $W(u)$ via the dual frame. Obviously, for a Parseval frame we have that

$$W^*W = I \quad \text{and} \quad u = \sum_{i=1}^{\infty} \langle u, u_k\rangle u_k.$$

It is important to note that an orthonormal basis of H is a Parseval frame but not conversely. If all elements of a Parseval frame have norm one then they constitute an orthonormal basis of H, see [82, Proposition 3.2.1]. In fact, the *redundancy* makes frames more useful than orthonormal bases for many image processing tasks.

In the discrete setting, where our images are represented as vectors in $\mathbb{R}^N$, we can write the analysis operator simply as a matrix $W \in \mathbb{R}^{M,N}$ with $M \geq N$, i.e., for a vector $u \in \mathbb{R}^N$ we have $(\langle u, u_k\rangle)_k = Wu$. The corresponding synthesis operator is then given by W^*. Clearly, the rows of W are the frame vectors. For Parseval frames we have $W^*W = I$ but in general it does not hold that $WW^* = I$.

A.2 Framelets

A popular way to construct frames is by means of wavelets. Roughly speaking, the key idea of wavelet theory is to work with an orthogonal basis of $L_2(\mathbb{R}^2)$ (or some other Hilbert space of functions) consisting of translated and dilated versions of a so-called *wavelet function* ψ. The dilations allow us to analyse and manipulate different "scales" of the given function. In the following, we will restrict our attention to functions in $L_2(\mathbb{R})$. The simplest extension to $L_2(\mathbb{R}^2)$ is given by the tensor product approach.

Let us start with the Haar wavelet function ψ in $L_2(\mathbb{R})$ which can be traced back to [125] and is defined as follows

$$\psi(x) = \begin{cases} 1 & \text{if } x \in [0, \frac{1}{2}), \\ -1 & \text{if } x \in [\frac{1}{2}, 1), \\ 0 & \text{otherwise,} \end{cases} \tag{A.4}$$

see also Fig. A.1 (right). The set of *dyadic dilations and translations* of ψ given by

$$\{\psi^{(j,k)} := 2^{-j/2}\psi(2^{-j}\cdot - k) \,:\, j,k \in \mathbb{Z}\},$$

constitutes an orthonormal basis of $L_2(\mathbb{R})$. In the following, we want to consider Parseval frames which are built upon wavelet functions. We will refer to them as *framelets*. Based on the ideas of [186], we will now show how framelets can be constructed by refinable functions. A function $\varphi \in L_2(\mathbb{R})$ is called a *refinable function* or a *scaling function*, if there exists a periodic function $H_\varphi \in L_\infty([-\frac{1}{2},\frac{1}{2}])$ such that it holds for the *Fourier transform* $\hat{\varphi}$ of φ that

$$\hat{\varphi}(2\omega) = H_\varphi(\omega)\hat{\varphi}(\omega). \tag{A.5}$$

We refer to (A.5) as the refinement equation or the two-scale relation for φ. The function H_φ is called the (refinement) *mask* of φ. Since we assume that H_φ is 1-periodic, we can consider the following expansion of H_φ

$$H_\varphi(w) = \frac{1}{2}\sum_{k\in\mathbb{Z}} h_\varphi[k]e^{-2\pi ik\omega},$$

so that (A.5) has the form

$$\hat{\varphi}(2\omega) = \frac{1}{2}\sum_{k\in\mathbb{Z}} h_\varphi[k]e^{-2\pi ik\omega}\hat{\varphi}(\omega) \tag{A.6}$$

for the coefficient sequence $h_\varphi = (h_\varphi[k])_{k\in\mathbb{Z}}$. From this coefficient sequence we immediately obtain the relation between two scales of φ in the image domain since it follows from (A.6) that

$$\varphi(x) = \sum_{k\in\mathbb{Z}} h_\varphi[k]\varphi(2x-k). \tag{A.7}$$

Orthonormal scaling functions define a so-called *multiresolution analysis* (MRA) of $L_2(\mathbb{R})$ which dates back to [151]. It consists of a sequence $(V_j)_{j\in\mathbb{Z}}$ of closed subspaces of $L_2(\mathbb{R})$ with the properties that

i) $\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots$,

ii) $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L_2(\mathbb{R}), \quad \bigcap_{j\in\mathbb{Z}} V_j = \{0\}$,

iii) $u \in V_j \Leftrightarrow u(\cdot - 2^j k) \in V_j, \quad \forall u \in L_2(\mathbb{R}),\ \forall j,k \in \mathbb{Z}$,

iv) $u \in V_j \Leftrightarrow u(2^j\cdot) \in V_0$,

v) $\{\varphi(\cdot - k)\,,\, k \in \mathbb{Z}\}$ is an orthonormal basis of V_0.

In view of i), let us define the sequence of spaces $(W_j)_{j\in\mathbb{Z}}$ by means of $W_j \oplus V_j = V_{j-1}$. It can be shown that

$$L_2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} W_j$$

and that there exists a wavelet function $\psi \in W_0$ whose integer translates form an orthonormal basis of W_0. Hence, it follows that $\{\psi^{(j,k)}\,,\, j,k \in \mathbb{Z}\}$ is an orthonormal basis of $L_2(\mathbb{R})$. More details can be found, e.g., in [71, 82, 152].

Since we are interested in Parseval frames and not orthonormal bases, we can work in a more general setting. We will now state the so-called *unitary extension principle* which was first proved in [186].

Theorem A.1 *Let $\varphi \in L_2(\mathbb{R})$ be a refinable function in the sense of* (A.5) *with respect to H_φ and suppose $\hat{\varphi}$ satisfies $\lim_{\omega\to 0}\hat{\varphi}(\omega) = 1$. Assume that $H_{\psi_1}, \ldots, H_{\psi_n}$ are periodic functions in $L_\infty([-\frac{1}{2},\frac{1}{2}])$ which satisfy*

$$H(\omega)^* H(\omega) = I, \quad \text{for a.e. } \omega \in [-\tfrac{1}{2},\tfrac{1}{2}]\ ,$$

where

$$H(\omega) := \begin{pmatrix} H_\varphi(\omega) & H_\varphi(\omega+\pi) \\ H_{\psi_1}(\omega) & H_{\psi_1}(\omega+\pi) \\ \vdots & \vdots \\ H_{\psi_n}(\omega) & H_{\psi_n}(\omega+\pi) \end{pmatrix}.$$

Let the wavelet functions ψ_l, $l = 1, \ldots, n$, be given by

$$\hat{\psi}_i(\omega) = H_{\psi_i}(\omega)\hat{\varphi}(\omega). \tag{A.8}$$

Then, the set of dilations and translations of the wavelet functions

$$\{\psi_l^{(j,k)} := 2^{-j/2}\psi_l(2^{-j}\cdot - k)\,,\, l = 1,\ldots,n,\, j,k \in \mathbb{Z}\}$$

is a Parseval frame of $L_2(\mathbb{R})$.

For more details on this result, we refer to [21, 33, 71, 84, 91, 186]. Clearly, we could have also formulated the unitary extension principle in terms the refinement masks and (A.6). A further generalization of the unitary extension principle which yields higher vanishing moments for the functions ψ_l, $l = 1, \ldots, n$, is the *oblique extension principle*, see [71, 84].

The easiest example of a pair of functions φ and ψ which fulfills the unitary extension principle is the Haar scaling function

$$\varphi(x) = \begin{cases} 1 & \text{if } x \in [0,1), \\ 0 & \text{otherwise} \end{cases}$$

and the Haar wavelet (A.4) which are depicted in Fig A.1. A short calculation shows that

$$H_\varphi(\omega) = \frac{1}{2}\big(1 + e^{-2\pi i w}\big), \quad H_\psi(\omega) = \frac{1}{2}\big(1 - e^{-2\pi i w}\big)$$

and we have the following coefficient sequences of the refinement mask

$$h_\varphi = [\,h_\varphi[0]\ \ h_\varphi[1]\,] = [1\ \ 1], \qquad h_\psi = [h_\psi[0]\ \ h_\psi[1]] = [1\ \ -1],$$

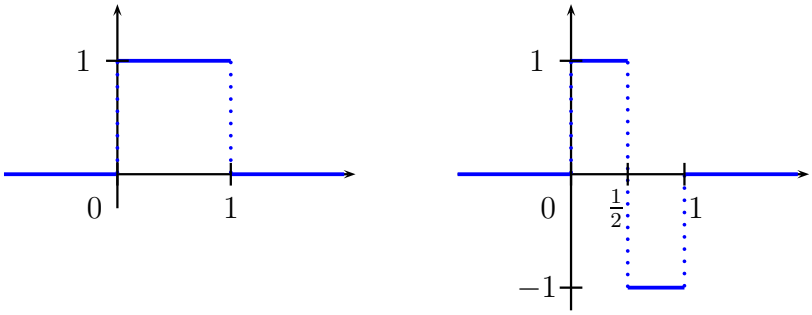


Figure A.1: The Haar scaling function φ (left) and the corresponding Haar wavelet function ψ on the right.

which are geometrically obvious regarding (A.7). Observe that if we translate the Haar scaling function by $-1/2$ we obtain the centered cardinal B-spline $B_1 = \chi_{[-1/2,1/2)}$. Its integer translates form a basis of the spline space of piecewise constant functions on $\mathbb{R}$ with knots at $\mathbb{Z} + \frac{1}{2}$. Higher-order spline subspaces of $C^{k-1}(\mathbb{R})$ of piecewise polynomial functions of degree k at most are spanned by integer translates of the B-splines B_{k+1}. It holds that these B-splines are convolutions of the B-spline B_1, i.e., $B_{k+1} = B_k * B_1$ for $k > 1$. Choosing these B-splines as scaling functions yields a classical example of the application of the unitary extension principle. The elements of the corresponding Parseval frame are called *B-spline framelets*. Let us restrict our attention to even-order B-splines. It holds for $\varphi = B_{2m}$ that

$$\hat{\varphi}(\omega) = \left(\frac{\sin(\pi\omega)}{\pi\omega}\right)^{2m}$$

and

$$H_\varphi(\omega) = \cos^{2m}(\pi\omega).$$

Together with $2m$ wavelet functions ψ_l, $l = 1, \ldots, 2m$, which are characterized by

$$H_{\psi_l}(\omega) = i^l \sqrt{\binom{2m}{l}} \sin^l(\pi\omega) \cos^{2m-l}(\pi\omega), \quad l = 1, \ldots, 2m, \tag{A.9}$$

the unitary extension principle is fulfilled, cf. [71]. For $m = 1$ the scaling function $\varphi = B_2$ and the two wavelet functions ψ_1 and ψ_2 are depicted in Fig. A.2. As we can also see geometrically from Fig. A.2, the corresponding coefficient sequences are given by

$$\begin{aligned}
h_\varphi &= [\, h_\varphi[-1] \;\; h_\varphi[0] \;\; h_\varphi[1] \,] = \frac{1}{2}[1 \;\; 2 \;\; 1], \\
h_{\psi_1} &= [\, h_{\psi_1}[-1] \;\; h_{\psi_1}[0] \;\; h_{\psi_1}[1] \,] = \frac{1}{\sqrt{2}}[1 \;\; 0 \;\; -1], \\
h_{\psi_2} &= [\, h_{\psi_2}[-1] \;\; h_{\psi_2}[0] \;\; h_{\psi_2}[1] \,] = \frac{1}{2}[1 \;\; -2 \;\; 1].
\end{aligned}$$

Observe that the integer translates of the functions φ, ψ_1 and ψ_2 are not orthogonal to each other. Consequently, the Parseval frame

$$\{\psi_1^{(j,k)} \,:\, j,k \in \mathbb{Z}\} \cup \{\psi_2^{(j,k)} \,:\, j,k \in \mathbb{Z}\}$$

which we obtain from the unitary extension principle of Theorem A.1 is, in contrast to the Haar case, not an orthonormal basis.

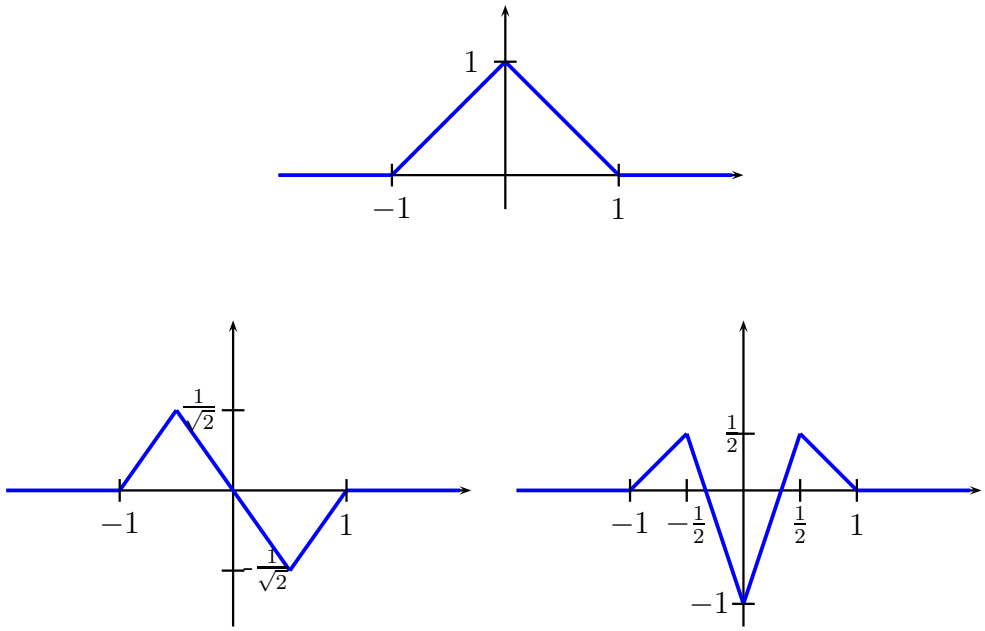


Figure A.2: Linear B-spline B_2 as scaling function (top) and the corresponding two wavelet functions ψ_1 and ψ_2 (bottom) which are characterized via (A.9) and (A.8). Note that the same results hold true if we use as wavelets the functions which result from reflecting ψ_1 and ψ_2 in the origin, cf. [51, 186].

We focus on the discrete setting now, i.e., instead of functions of $L_2(\mathbb{R})$ we consider vectors of $\mathbb{R}^n$. For the sake of simplicity, we assume that d is even. Clearly, corresponding framelet transformations can only by defined up to the one-pixel level which means that the discrete counterpart of our framelets must always contain part of the scaling function. Furthermore, the discrete setting means that we can simply build a Parseval frame matrix by using the refinement masks in the rows of the frame matrix as we will see below.

We consider now the Haar scaling function with coefficient sequence $[1\ \ 1]$ and the linear B-spline scaling function with coefficient sequence $\frac{1}{2}[1\ \ 2\ \ 1]$. Let us restrict our attention to a single decomposition level and let us use periodic boundary conditions. Then, the unitary extension principle and in the Haar case already the classical MRA yields Parseval frames of the form

$$W = \begin{pmatrix} H_0 \\ H_1 \end{pmatrix}, \tag{A.10}$$

where the submatrix $H_0 \in R^{n,n}$ originates from the scaled coefficient sequence of the scaling function and $H_1 \in R^{n,n}$ from the corresponding scaled coefficient sequence for the wavelet functions, i.e.,

$$H_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & & & & \\ & & 1 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & 1 \end{pmatrix}, \quad H_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & & & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -1 \end{pmatrix}.$$

Observe that the matrix W obtained by (A.10) is orthogonal since the scaling function themselves are orthonormal and there is no redundancy. In contrast to that, the translated scaling functions of the linear B-spline overlap. We thus get a genuine Parseval frame of the form (A.10) with

$$H_0 = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & 2 & 1 & \\ 1 & & & & & & 1 & 2 \end{pmatrix}$$

and $H_1 = \begin{pmatrix} H_{1a} \\ H_{1b} \end{pmatrix}$ where

$$H_{1a} = \frac{1}{2} \begin{pmatrix} 1 & 0 & -1 & & & & & \\ & & 1 & 0 & -1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & 0 & -1 & \\ -1 & & & & & & 1 & 0 \end{pmatrix},$$

$$H_{1b} = \frac{1}{\sqrt{8}} \begin{pmatrix} 1 & -2 & 1 & & & & & \\ & & 1 & -2 & 1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 1 & -2 & 1 & \\ 1 & & & & & & 1 & -2 \end{pmatrix}.$$

For both the Haar and the B-spline case there is a straightforward way to increase the redundancy by a factor of two by omitting the downsampling operation. This is a well-known technique in the context of *filter banks*, see, e.g., [214] and for applications, e.g., [175].

In our case, the framelets we obtain without downsampling are sometimes referred to as *undecimated* Haar or B-spline framelets. The corresponding components of the Parseval frame matrix are simply the $n \times n$ circulant matrices of the scaled coefficient sequences of

the above masks. In the Haar case, we get

$$H_0 = \frac{1}{2}\begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{pmatrix}, \quad H_1 = \frac{1}{2}\begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ -1 & & & & 1 \end{pmatrix}. \tag{A.11}$$

The matrix

$$W = \begin{pmatrix} H_0 \\ H_1 \end{pmatrix} \tag{A.12}$$

is a Parseval frame matrix but not an orthogonal matrix anymore. Similarly, we can construct for the linear B-spline the Parseval frame matrix $W = \begin{pmatrix} H_0 \\ H_{1a} \\ H_{1b} \end{pmatrix}$ by

$$H_0 = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 & & & \\ & 1 & 2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 2 & 1 \\ 1 & & & & 1 & 2 \\ 2 & 1 & & & & 1 \end{pmatrix}$$

and

$$H_{1a} = \frac{1}{\sqrt{8}}\begin{pmatrix} 1 & 0 & -1 & & & & \\ & 1 & 0 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 0 & -1 & \\ -1 & & & & 1 & 0 & \\ 0 & -1 & & & & 1 & \end{pmatrix}, \quad H_{1b} = \frac{1}{4}\begin{pmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ 1 & & & & 1 & -2 & \\ -2 & 1 & & & & 1 & \end{pmatrix}.$$

We can get similar Parseval frames for higher order B-spline wavelets and other boundary condition, e.g., mirrored/Neumann boundary conditions are possible, cf. [51]. Recall that, so far, we have considered just the finest level on which we can, in the discrete case, define our scaling and wavelet functions. Coarser levels, on the other hand, are possible. To simplify the notation, we focus on the case of undecimated Haar framelets. The construction for multi-level B-spline framelets works analogously. Let us fix a decomposition level L ($L < n$ in the Haar case). Again, we do not downsample so that we consider for $l = 0, \ldots, L-1$ the circulant $n \times n$ matrices $H_0^{(l)}$ and $H_1^{(l)}$ with respect to the filters

$$\frac{1}{2}[\,1 \ \underbrace{0 \ \ldots \ 0}_{2^{l-1}-1} \ (-1)^i\,], \quad i = 1, 2.$$

Then, the undecimated Haar framelet matrix up to level L is given by

$$W = \begin{pmatrix} \Pi_{l=0}^{L-1} H_0^{(L-l)} \\ H_1^{(1)} \\ H_1^{(2)} H_0^{(1)} \\ \vdots \\ H_1^{(L-1)} \Pi_{l=2}^{L-1} H_0^{(L-l)} \\ H_1^{(L)} \Pi_{l=1}^{L-1} H_0^{(L-l)} \end{pmatrix}.$$

So far, we have focused on the one-dimensional case. Parseval frames for digital 2D images can be obtained by a simple tensor product construction. For example, we get for one level of the Haar Parseval frame (A.12) the following frame for digital $n \times n$ images reshaped into a vector of length $N = n^2$:

$$W = \begin{pmatrix} H_0 \otimes H_0 \\ H_0 \otimes H_1 \\ H_1 \otimes H_0 \\ H_1 \otimes H_1 \end{pmatrix}. \tag{A.13}$$

The rows of W form again a Parseval frame, i.e, W fulfills $W^{\mathrm{T}}W = I_N$ but $W^{\mathrm{T}}W \neq I_{4N}$. In the same way, we can construct the corresponding Haar and B-spline framelets in the multi-level setting for images.

APPENDIX B

Matrix diagonalization via cosine transform

Let us consider matrices $A \in \mathbb{R}^{n,n}$ of the form

$$A = \text{stoep}(a_0, \ldots, a_{n-1}) + \text{pshank}(a_1, \ldots, a_{n-2}, 0), \tag{B.1}$$

where $\text{stoep}(a_0, \ldots, a_{n-1})$ is a symmetric Toeplitz matrix and $\text{pshank}(a_1, \ldots, a_{n-2}, 0)$ is a persymmetric Hankel matrix, i.e.,

$$\text{stoep}(a_0, \ldots, a_{n-1}) := \begin{pmatrix} a_0 & a_1 & \ldots & a_{n-2} & a_{n-1} \\ a_1 & a_0 & \ldots & a_{n-3} & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & a_{n-3} & \ldots & a_0 & a_1 \\ a_{n-1} & a_{n-2} & \ldots & a_1 & a_0 \end{pmatrix},$$

$$\text{pshank}(a_1, \ldots, a_{n-1}, 0) := \begin{pmatrix} a_1 & a_2 & \ldots & a_{n-2} & 0 \\ a_2 & a_3 & \ldots & 0 & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-2} & 0 & \ldots & a_3 & a_3 \\ 0 & a_{n-2} & \ldots & a_2 & a_1 \end{pmatrix}.$$

The following result shows that matrices with the structure (B.1) can be diagonalized via the matrix of the cosine-II transform given by

$$C_{II} := \left(\frac{2}{n}\right)^{1/2} \left(\varepsilon_j \cos \frac{j(2k+1)\pi}{2n}\right)_{j,k=0}^{n-1} \in \mathbb{R}^{n,n}$$

with $\varepsilon_0 := 1/\sqrt{2}$ and $\varepsilon_j := 1$ for $j \neq 0$.

Proposition B.1 *Let $A \in \mathbb{R}^{n,n}$ be a matrix of the form* (B.1)*. Then, it holds that*

$$A = C_{II}^T \, \text{diag}(q) \, C_{II},$$

where

$$q_j = 2 \sum_{k=0}^{n-1} \varepsilon_k^2 a_k \cos \frac{jk\pi}{n}, \quad j = 0, \ldots, n-1.$$

Proof: See, e.g., [178].

For the forward difference matrix D_f the matrix $D_f^{\mathrm{T}} D_f$ is of the form (B.1) with $(a_0, \ldots, a_{n-1}) = (2, -1, 0, \ldots, 0)$. Hence, we can apply Proposition B.1 and a short calculation shows that the vector $q \in \mathbb{R}^n$ is in this case given by

$$q_j = \big(2 \sin \frac{\pi j}{2n}\big)^2, \quad j = 0, \ldots, n-1.$$

It follows for the matrix D_2 defined in (3.18) that

$$\begin{aligned} D_2^{\mathrm{T}} D_2 &= I_n \otimes D_f^{\mathrm{T}} D_f + D_f^{\mathrm{T}} D_f \otimes I_n \\ &= (C_{II}^{\mathrm{T}} \otimes C_{II}^{\mathrm{T}})(I_n \otimes \mathrm{diag}(q) + \mathrm{diag}(q) \otimes I_n)(C_{II} \otimes C_{II}). \end{aligned} \tag{B.2}$$

We can use (B.2), e.g., to apply the inverse matrix $(\gamma I + D_2^{\mathrm{T}} D_2)^{-1}$ in $\mathcal{O}(n^2 \log n)$ steps since the multiplications of an $n \times n$ matrix by the cosine transform matrices C_{II} can be implemented using fast cosine or FFT algorithms.

APPENDIX C

Chambolle's semi-implicit gradient descent algorithm

In [56], Chambolle proposed the following algorithm to solve the dual of the discrete ROF denoising problem, i.e., problem (3.26) with $\Phi = \Phi_2$ and $D = D_2$:

Algorithm (Semi-implicit gradient descent algorithm)
Initialization: $b^{(0)}$
For $k = 0, 1, \ldots$ repeat until a stopping criterion is reached

$$\begin{aligned} w^{(k+1)} &:= D(D^{\mathrm{T}} b^{(k)} - f), \\ b^{(k+1)} &:= \left(1 + \frac{\gamma}{\lambda}\begin{pmatrix} |w^{(k+1)}| \\ |w^{(k+1)}| \end{pmatrix}\right)^{-1} \circ (b^{(k)} + \gamma w^{(k)}), \end{aligned} \tag{C.1}$$

where $|w^{(k+1)}|$ is defined as in (3.19).

Recall that the solution of the primal problem (3.12) is then given by $\hat{u} = f - D^{\mathrm{T}}\hat{b}$. The inversion and multiplication in the second step of (C.1) is again meant componentwise. Clearly, algorithm (C.1) can also be used to solve the Besov-norm denoising problem. In this case, we have to use the componentwise absolute value of $w^{(k+1)}$ in the second step of the above algorithm instead of $\begin{pmatrix} |w^{(k+1)}| \\ |w^{(k+1)}| \end{pmatrix}$. The algorithm (C.1) can be motivated by using a semi-implicit scheme on the corresponding Karush-Kuhn-Tucker conditions, see also [200]. So far, convergence was only proved for the case $\gamma < \frac{1}{\|D^{\mathrm{T}}D\|_2}$ and $b^{(0)}$ with the property that $|b^{(0)}| \leq \lambda$. However, in all numerical experiments this algorithm converged for step sizes $\gamma < \frac{2}{\|D^{\mathrm{T}}D\|_2}$. Recall that the related forward-backward splitting algorithm is guaranteed to converge for step lengths $\gamma < \frac{2}{\|D^{\mathrm{T}}D\|_2}$ and all initial values.

In the following, we want to examine the "cumbersome" behavior of Chambolle's algorithm in *one variable*. Here, problem (3.26) is given by

$$\operatorname*{argmin}_{b \in \mathbb{R}} \frac{1}{2}(db - f)^2 \quad \text{s.t.} \quad |b| \leq \lambda, \tag{C.2}$$

where we assume that $d > 0$ is a real number. The solution $\hat{b}$ of (C.2) is given by

$$\begin{aligned} \hat{b} &= \begin{cases} \frac{f}{d} & \text{if } |\frac{f}{d}| \le \lambda, \\ \lambda & \text{if } \frac{f}{d} > \lambda, \\ -\lambda & \text{if } \frac{f}{d} < -\lambda, \end{cases} \\ &= \frac{f}{d} - \mathcal{T}_\lambda(\tfrac{f}{d}). \end{aligned} \tag{C.3}$$

The corresponding forward-backward splitting algorithm, cf., (3.33) has the form

$$b^{(k+1)} = \frac{b^{(k)} - \gamma d(db^{(k)} - f)}{\max\{1, \frac{1}{\lambda}(b^{(k)} - \gamma d(db^{(k)} - f))\}}. \tag{C.4}$$

Let us view this as a dynamical system, i.e., we define the fixed point operator $T_{FBS} : \mathbb{R} \to \mathbb{R}$ by

$$T_{FBS}(b) = \frac{b - \gamma d(db - f)}{\max\{1, \frac{1}{\lambda}(b^{(k)} - \gamma d(db^{(k)} - f))\}}$$

and consider Picard iterations $b^{(k+1)} = T_{FBS}(b^{(k)})$.

For the semi-implicit gradient descent algorithm (C.1) we obtain

$$b^{(k+1)} = \frac{b^{(k)} - \gamma d(db^{(k)} - f)}{1 + \frac{\gamma}{\lambda}|d(db^{(k)} - f)|} \tag{C.5}$$

and we define as above the fixed point operator $T_{Ch} : \mathbb{R} \to \mathbb{R}$ by

$$T_{Ch}(b) = \frac{b - \gamma d(db - f)}{1 + \frac{\gamma}{\lambda}|d(db - f)|}. \tag{C.6}$$

To analyse the corresponding sequences we borrow a tool from the theory of dynamical systems, namely, the visualization techniques of *cobweb diagrams*, see, e.g., [36]. We already know that the forward-backward splitting algorithm (C.4) converges for every $\gamma < \frac{2}{d^2}$ and every starting value $b^{(0)}$ to the solution $\hat{b}$ which thus must be the unique fixed point of T_{FBS}, cf., Fig C.1. In the case of (C.6), a fixed point $\tilde{b}$ is characterized by

$$\tilde{b} = T_{Ch}(\tilde{b}) \quad \Leftrightarrow \quad \frac{\tilde{b}}{\lambda}|d\tilde{b} - f| = -(d\tilde{b} - f).$$

Hence, the fixed points of T_{Ch} are given by

$$\tilde{b}_1 = \frac{f}{d} \quad \text{and} \quad \begin{aligned} \tilde{b}_2 &= \lambda \quad \text{if } \lambda < \tfrac{f}{d}, \\ \tilde{b}_2 &= -\lambda \quad \text{if } \lambda < -\tfrac{f}{d}. \end{aligned} \tag{C.7}$$

The function T_{Ch} is differentiable everywhere except at $\tilde{b}_1 = \frac{f}{d}$ where the left and right derivatives are given by

$$T'_{Ch}\big((\tfrac{f}{d})_+\big) = 1 - \gamma\big(\tfrac{df}{\lambda} + d^2\big) \tag{C.8}$$

and

$$T'_{Ch}\big((\tfrac{f}{d})_-\big) = 1 + \gamma\big(\tfrac{df}{\lambda} - d^2\big). \tag{C.9}$$

Without loss of generality, let us study the case $f \ge 0$ in more detail. By (C.7), we have to distinguish between the following two situations.

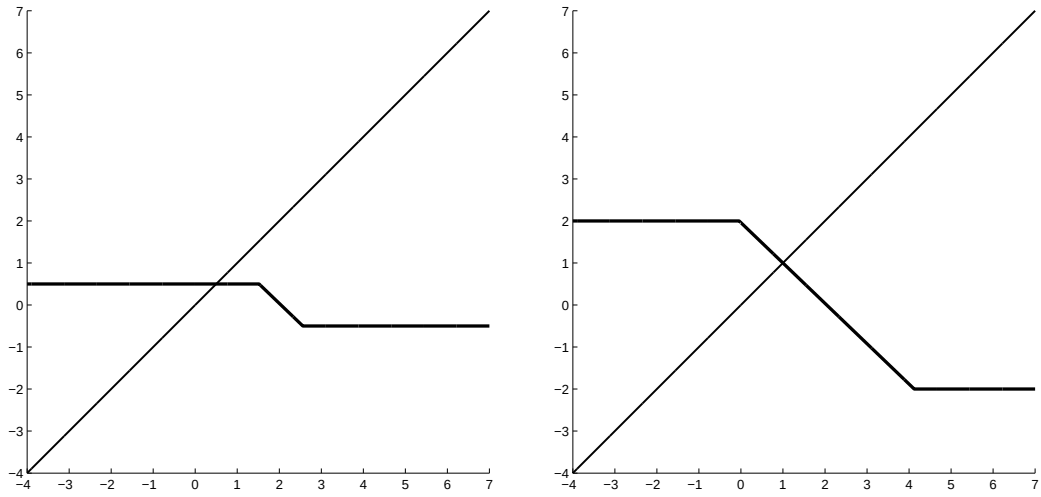


Figure C.1: Cobweb diagrams of the function T_{FBS} for $d = 2, f = 2, \gamma = 0.49$. On the left-hand side, $\lambda = 0.5$ and the constraint is active, i.e., $\hat{b} = 0.5$, whereas on the right-hand side $\lambda = 2$ which means that $\hat{b} = 1$.

i) $\lambda < \frac{f}{d}$
According to (C.7), we have two distinct fixed points, $\tilde{b}_1 = \frac{f}{d}$ and $\tilde{b}_2 = \hat{b} = \lambda$. Clearly, it holds that $T'_{Ch}\big((\frac{f}{d})_-\big) > 1$ for all $\gamma > 0$. The right derivative, in contrast, vanishes for

$$\gamma^* = \frac{1}{\frac{df}{\lambda} + d^2}.$$

If $\gamma > \gamma^*$, we have that $T'_{Ch}\big((\frac{f}{d})_+\big) < 0$ and thus the corresponding cobweb has the form shown on the left-hand side of Fig. C.2. We see that $\tilde{b}_1 = \frac{f}{d}$ is a *repelling fixed point*. In contrast, $\hat{b}_2 = \lambda$ is an *attracting fixed point* since its derivative lies in $(0, 1)$. We have shown the three possible scenarios for different starting values in this diagram: $b^{(0)} \leq \tilde{b}_2$ (green), $\tilde{b}_2 \leq b^{(0)} < \tilde{b}_1$ (red) and $b^{(0)} > \tilde{b}_1$ (blue). The sequence generated by (C.5) converges to $\tilde{b}_2 = \hat{b}$, i.e., to the solution of problem (C.2), for each initial value $b^{(0)}$ except for the case where $b^{(0)} = \tilde{b}_1$.

The diagram for the case $\gamma \leq \gamma^*$ has the structure shown on the right-hand side of Fig. C.2: $(b^{(k)})_{k\in\mathbb{N}}$ converges to $\tilde{b}_2 = \lambda$ if $b^{(0)} < \frac{f}{d}$. In the case $b^{(0)} \geq \frac{f}{d}$ we have convergence to $\tilde{b}_1$ which is not the solution of (C.2).

ii) $\lambda \geq \frac{f}{d}$
By (C.3), we have only one fixed point $\tilde{b}_1 = \hat{b} = \frac{f}{d}$. It follows from the definition of the right and left derivatives, cf., (C.8) and (C.9), that $|T'_{Ch}\big((\frac{f}{d})_+\big)| < 1$ if and only if $\gamma < \frac{2}{d^2 + \frac{df}{\lambda}}$ and $|T'_{Ch}\big((\frac{f}{d})_-\big)| < 1$ if and only if $\gamma < \frac{2}{d^2 - \frac{df}{\lambda}}$. So, for $\gamma < \frac{2}{d^2 + \frac{df}{\lambda}}$ we have that $\tilde{b}_1 = \frac{f}{d}$ is an attracting fixed point, see the diagram on the left-hand side of Fig.

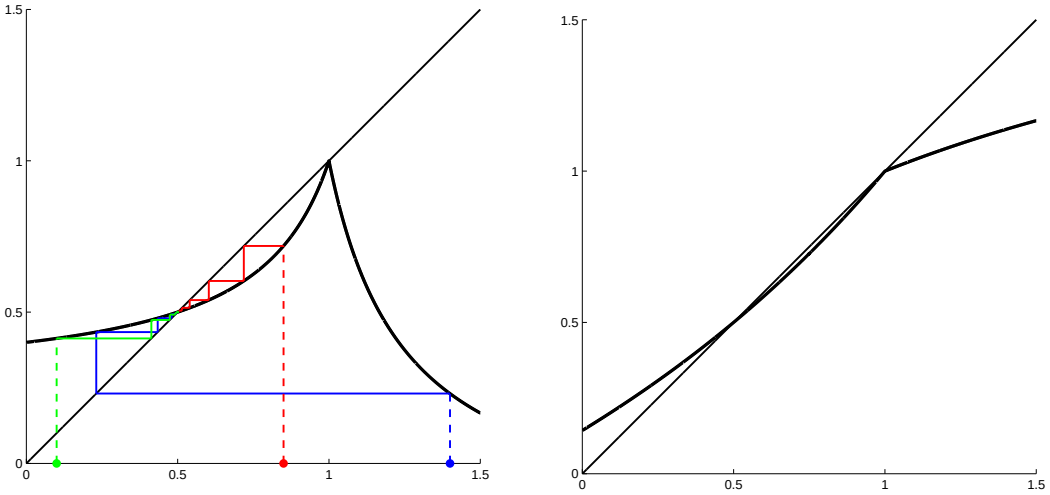


Figure C.2: Cobweb diagrams of the function T_{Ch} for the case $d = 2, f = 2, \lambda = 0.5$. The fixed points are $\tilde{b}_1 = 1$ and $\tilde{b}_2 = 0.5$ and we have $\gamma^* = 0.08\bar{3}$. Left: $\gamma = 0.5 > \gamma^*$. Left: $\gamma = 0.5$. Right: $\gamma = 0.05 < \gamma^*$.

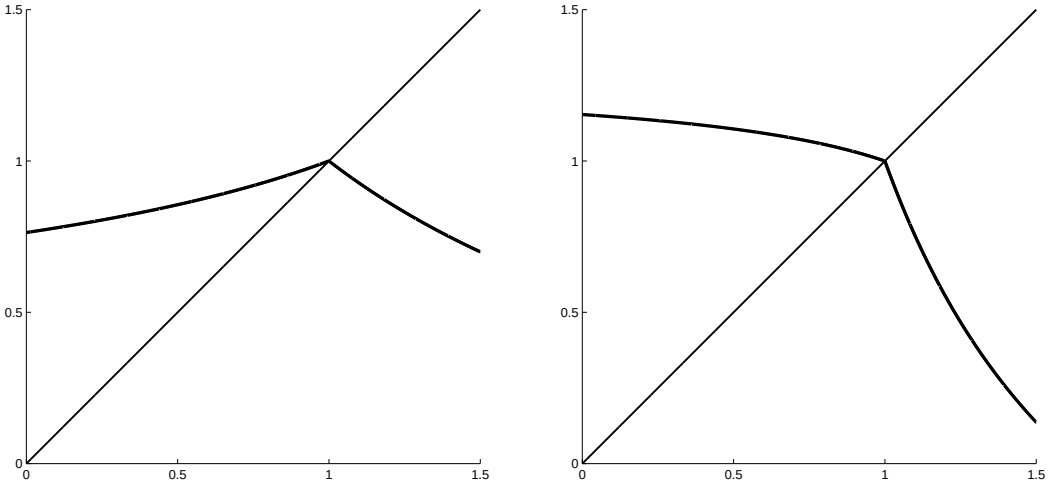


Figure C.3: The function T_{Ch} and the identity mapping in the case $d = 2, f = 2, \lambda = 2$ and $\gamma = 0.3$ (left) and $\gamma = 0.64$ (right). Note that $\frac{2}{d^2 + \frac{df}{\lambda}} \approx 0.3387$ and $\frac{2}{d^2 - \frac{f^2}{\lambda^2}} \approx 0.6466$.

C.3. However, we will show below that this condition can be weakened, in particular, we can guarantee convergence for

$$\gamma < \frac{2}{d^2 - \frac{f^2}{\lambda^2}}.$$

Let us assume from now on that $\gamma \geq \frac{2}{d^2+\frac{df}{\lambda}}$ which implies that $T'_{Ch}\big((\frac{f}{d})_+\big) < -1$. First, we consider the case where $\frac{1}{d^2-\frac{df}{\lambda}} \leq \frac{2}{d^2+\frac{df}{\lambda}}$, or equivalently, $\lambda \geq \frac{3f}{d}$. In this situation, $T'_{Ch}\big((\frac{f}{d})_-\big) \leq 0$ and $\tilde{b}_1$ is an attracting fixed point if and only if

$$T'_{Ch}\big((\tfrac{f}{d})_+\big)T'_{Ch}\big((\tfrac{b}{d})_-\big) < 1.$$

We have

$$\begin{aligned} T'_{Ch}\big((\tfrac{f}{d})_+\big)T'_{Ch}\big((\tfrac{f}{d})_-\big) &= \big(1-\gamma\big(\tfrac{df}{\lambda}+d^2\big)\big)\big(1+\gamma\big(\tfrac{df}{\lambda}-d^2\big)\big) \\ &= 1-\gamma d^2\big(2-\gamma\big(d^2-\tfrac{f^2}{\lambda^2}\big)\big) \end{aligned}$$

and obtain

$$T'_{Ch}\big((\tfrac{f}{d})_+\big)T'_{Ch}\big((\tfrac{f}{d})_-\big) < 1 \quad \Leftrightarrow \quad \gamma < \frac{2}{d^2 - \frac{f^2}{\lambda^2}}.$$

Second, we consider the case $\frac{1}{d^2-\frac{df}{\lambda}} > \frac{2}{d^2+\frac{df}{\lambda}}$, i.e., $\frac{3f}{d} > \lambda \geq \frac{f}{d}$. We apply the same argument as above to conclude convergence for $\gamma \in [\frac{1}{d^2-\frac{df}{\lambda}}, \frac{2}{d^2-\frac{f^2}{\lambda^2}})$ since here $T'_{Ch}\big((\frac{f}{d})_-\big) \leq 0$. For the remaining case $\gamma \in [\frac{2}{d^2+\frac{df}{\lambda}}, \frac{1}{d^2-\frac{df}{\lambda}})$ convergence can be verified using that $T'_{Ch}\big((\frac{b}{d})_-\big) \in (0,1)$ and $T'_{Ch}\big((\frac{b}{d})_+\big) < 0$.

The following Table C.1 summarizes the cases where we have convergence to the minimizer of (C.2). Note that here we still assume that $d > 0$ and $f \geq 0$. We obtain the corresponding result for $f \leq 0$ if we replace f by $-f$ in the second and the third column and flipping the inequality sign in the fourth column.

Case	λ	γ	$b^{(0)}$
1	$\lambda < \frac{f}{d}$	$\gamma > \frac{1}{\frac{df}{\lambda}+d^2}$	$b^{(0)} \neq \frac{f}{d}$
2	$\lambda < \frac{f}{d}$	$\gamma \leq \frac{1}{\frac{df}{\lambda}+d^2}$	$b^{(0)} < \frac{f}{d}$
3	$\lambda \geq \frac{f}{d}$	$\gamma < \frac{2}{d^2-\frac{f^2}{\lambda^2}}$	$b^{(0)} \in \mathbb{R}$
4	$\lambda \geq \frac{f}{d}$	$\gamma \in \mathbb{R}$	$b^{(0)} = \frac{f}{d}$

Table C.1: Conditions on λ, γ and $b^{(0)}$ which guarantee convergence of the sequence generated by (C.5) to $\hat{b}$ for the case where $f \geq 0$ and $d > 0$.

Recall that in the $2D$ case, convergence is assumed to be true but not proven yet for the case $\gamma < \frac{2}{\|D\|^2}$ and $b^{(0)}$ fulfilling $|b^{(0)}| \leq \lambda$. We see from Table C.1 that at least in this simple case of one variable, we have shown convergence under even weaker conditions.

Bibliography

[1] F. Acker and M.-A. Prestel. Convergence d'un schéma de minimisation alternée. *Annales de la Faculté des Sciences de Toulouse*, 2(1):1–9, 1980.

[2] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. Academic Press, Amsterdam, 2003.

[3] D. C. Alexander. Modelling, fitting and sampling in diffusion MRI. In D. H. Laidlaw and J. Weickert, editors, *Visualization and Processing of Tensor Fields, Advances and Perspectives*, Mathematics and Visualization, pages 3–20. Springer, Berlin, 2009.

[4] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95:3–51, 2003.

[5] G. Aubert and J.-F. Aujol. A variational approach to removing multiplicative noise. *SIAM Journal on Applied Mathematics*, 68(4):925–946, 2008.

[6] J.-P. Aubin. *Optima and Equilibria: An Introduction to Nonlinear Analysis*. Springer, Berlin, Heidelberg, New York, 2nd edition, 2003.

[7] J. P. Aubin and H. Frankowska. *Set-valued Analysis*. Birkäuser, Boston, 1990.

[8] J.-F. Aujol. Some first-order algorithms for total variation based image restoration. *Journal of Mathematical Imaging and Vision*, 34(3):307–327, 2009.

[9] J.-F. Aujol and A. Chambolle. Dual norms and image decomposition models. *International Journal of Computer Vision*, 63(1):71–88, 2005.

[10] J.-B. Baillon and G. Haddad. Quelques propriétés des opérateurs angle-bornés et n-cycliquement monotones. *Israel Journal of Mathematics*, 26(2):137–150, 1977.

[11] A. Barvinok. *A Course in Convexity, Graduate Studies in Mathematics*. AMS, Providence, RI, 2002.

[12] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[13] P. J. Basser. Inferring microstructural features and the physical state of tissues from diffusion-weighted images. *Nuclear Magnetic Resonance in Biomedicine*, 8:333–334, 1995.

[14] P. J. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. 66:259–267, 1994.

[15] S. Basu, T. Fletcher, and R. Whitaker. Rician noise removal in diffusion tensor MRI. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2006*, volume 4190 of *Lecture Notes in Computer Science*, pages 117–125. Springer, 2006.

[16] H. H. Bauschke and P. L. Combettes. The Baillon-Haddad theorem revisited. 2009. Preprint on http://arXiv.org/.

[17] H. H. Bauschke, P. L. Combettes, and S. Reich. The asymptotic behaviour of the composition of two resolvents. *Nonlinear Analysis*, 60:283–301, 2005.

[18] P. Bechler, R. DeVore, A. Kamont, G. Petrova, and P. Wojtaszczyk. Greedy wavelet projections are bounded on BV. *Transactions of the American Mathematical Society*, 359(2):619–635, 2006.

[19] A. Beck and M. Teboulle. Fast iterative shrinkage-thresholding algorithm for linear inverse problems. *IEEE Transactions on Image Processing*, 2008. Accepted for publication.

[20] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009.

[21] J. Benedetto and O. Treiber. Wavelet frames: multiresolution analysis and extension principles. In L. Debnath, editor, *Wavelet transforms and time-frequency signal analysis*, pages 1–36. Birkhäuser, Boston, 2001.

[22] B. Berkels, M. Burger, M. Droske, O. Nemitz, and M. Rumpf. Cartoon extraction based on anisotropic image classification. In L. Koppelt, T. Kuhlen, T. Aach, and R. Westermann, editors, *Vision, Modeling and Visualization 2006*, pages 293–300. IOS Press, Amsterdam, The Netherlands, 2006.

[23] A. Bermùdez and C. Moreno. Duality methods for solving variational inequalities. *Computers & Mathematics with Applications*, 7:43–58, 1981.

[24] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of SIGGRAPH*, pages 417–424, 2000.

[25] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture inpainting. *IEEE Transactions on Image Processing*, 12:882–889, 2003.

[26] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. IoP Publishing, Bristol, 1998.

[27] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York, 1982.

[28] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[29] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*, volume 7 of *Springer Series in Operations Research*. Springer, Berlin, 2000.

[30] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, New York, 2000.

[31] J. M. Borwein and Q. J. Zhu. *Techniques of variational analysis*. Springer, New York, 2005.

[32] J. M. Borwein and Q. J. Zhu. Variational methods in convex analysis. *Journal of Global Optimization*, 35(2):197–213, 2006.

[33] M. Bownik. Tight frames of multidimensional wavelets. *Journal of Fourier Analysis and Applications*, 3(5):525–542, 1997.

[34] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[35] H. Brezis. *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*. North–Holland, Amsterdam, 1973.

[36] M. Brin and G. Stuck. *Introduction to Dynamical Stystems*. Cambridge University Press, 2002.

[37] F. E. Browder and W. V. Petryshyn. The solution by iteration of nonlinear functional equations in Banach spaces. *Bulletin of the American Mathematical Society*, 72:571–575, 1966.

[38] T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. *Image and Vision Computing*, 24(1):41–55, 2006.

[39] C. Brune, A. Sawatzky, and M. Burger. Primal and dual Bregman methods with application to optical nanoscopy. 2009. Submitted to the International Journal of Computer Vision.

[40] C. Brune, A. Sawatzky, T. Kösters, F. Wübbeling, and M. Burger. An analytical view on EM-TV based methods for inverse problems with Poisson noise. *Preprint, University of Münster*, 2009.

[41] A. Buades, B. Coll, and J.-M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[42] B. Burgeth, A. Bruhn, S. Didas, J. Weickert, and M. Welk. Morphology for matrix-data: Ordering versus PDE-based approach. *Image and Vision Computing*, 25(4):496–511, 2007.

[43] B. Burgeth, A. Bruhn, N. Papenberg, M. Welk, and J. Weickert. Mathematical morphology for matrix fields induced by the Loewner ordering in higher dimensions. *Signal Processing*, 87(2):277–290, 2007.

[44] B. Burgeth, S. Didas, L. Florack, and J. Weickert. Singular PDEs for the processing of matrix-valued data. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 556–567. Springer, Berlin, 2007.

[45] B. Burgeth, S. Didas, and J. Weickert. A generic approach to diffusion filtering of matrix-fields. *Computing*, 81:179–197, 2007.

[46] D. Butnariu and A. N. Iusem. *Totally convex functions for fixed points computation and infinite dimensional optimization*, volume 40 of *Applied Optimization*. Kluwer, Dordrecht, 2000.

[47] C. Byrne. A unified treatment of some iterative algorithms in signal processing and image reconstruction. *Inverse Problems*, 20:103–120, 2004.

[48] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. Technical report, UCLA Computational and Applied Mathematics, 2008.

[49] J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen. Simultaneous inpainting in image and transformed domains. *Numerische Mathematik*, 112(4):509–533, 2009.

[50] J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen. Convergence analysis of tight framelet approach for missing data recovery. *Advances in Computational Mathematics*, in print, 2008.

[51] J.-F. Cai, R. H. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24:131–149, 2008.

[52] J.-F. Cai, S. Osher, and Z. Shen. Split Bregman methods and frame based image restoration. Technical report, UCLA Computational and Applied Mathematics, March 2009.

[53] Y. Censor and A. Lent. An iterative row-action method for interval convex programming. *Journal of Optimization Theory and Applications*, 34(3):321–353, 1981.

[54] Y. Censor and S. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, 1997.

[55] Y. Censor and S. A. Zenios. Proximal minimization algorithm with D-functions. *Journal of Optimization Theory and Applications*, 73(3):451–464, 1992.

[56] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

[57] A. Chambolle. Total variation minimization and a class of binary MRF models. In A. Rangarajan, B. C. Vemuri, and A. L. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition, EMMCVPR*, volume 3757 of *LNCS*, pages 136–152. Springer, 2005.

[58] A. Chambolle, R. A. DeVore, N. Lee, and B. L. Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3):319–335, 1998.

[59] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76:167–188, 1997.

[60] R. H. Chan, C.-W. Ho, and M. Nikolova. Salt-and-pepper noise removal by median noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*, 14(10), 2005.

[61] R. H. Chan, S. Setzer, and G. Steidl. Inpainting by flexible Haar-wavelet shrinkage. *SIAM Journal on Imaging Science*, 1:273–293, 2008.

[62] T. F. Chan, S. Esedoglu, and F. Park. A fourth order dual method for staircase reduction in texture extraction and image restoration problems. Technical report, UCLA Computational and Applied Mathematics, 2005.

[63] T. F. Chan, S. Esedoglu, and F. Park. Image decomposition combining staircase reduction and texture extraction. *Journal of Visual Communication and Image Representation*, 18(6):464–486, 2007.

[64] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal–dual method for total-variation based image restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977, 1999.

[65] T. F. Chan, S. H. Kang, and J. Shen. Euler's elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, 63:564–592, 2002.

[66] T. F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, 2005.

[67] C. Chaux, J.-C. Pesquet, and N. Pustelnik. Nested iterative algorithms for convex constrained image recovery problems. *SIAM Journal on Imaging Sciences*, 2(2):730–762, 2009.

[68] C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras. Constrained flows of matrix-valued functions: application to diffusion tensor regularization. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision – ECCV 2002*, volume 2350 of *Lecture Notes in Computer Science*, pages 251–265. Springer, Berlin, 2002.

[69] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[70] G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathematical Programming*, 64:81–101, 1994.

[71] O. Christensen. *An Introduction to Frames and Riesz Bases*. Applied and Numerical Harmonic Analysis. Birkhäuser, Boston, 2003.

[72] P. G. Ciarlet. *Introduction to Numerical Linear Algebra and Optimization*. Cambridge University Press, Cambridge, 1989.

[73] A. Cohen. *Numerical Analysis of Wavelet Methods*, volume 32 of *Studies in Mathematics and its Applications*. Elsevier, Amsterdam, 1988.

[74] A. Cohen, R. DeVore, P. Petrushev, and H. Xu. Nonlinear approximation and the space BV($\mathbb{R}^2$). *American Journal of Mathematics*, 121:587–628, 1999.

[75] P. L. Combettes. Quasi-Fejérian analysis of some optimization algorithms. In Y. C. D. Butnariu and S. Reich, editors, *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pages 115–152, Elsevier, New York, 2001.

[76] P. L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5–6):475–504, 2004.

[77] P. L. Combettes. Iterative construction of the resolvent of a sum of maximal monotone operators. *Journal of Convex Analysis*, 16(4), 2009.

[78] P. L. Combettes and J.-C. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574, 2007.

[79] P. L. Combettes and J.-C. Pesquet. A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, 24(6), 2008.

[80] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modelling & Simulation*, 4:1168–1200, 2005.

[81] I. Csiszár. Why least squares and maximum entropy? An automatic approach to inference for linear inverse problems. *The Annals of Statistics*, 19(4):2032–2066, 1991.

[82] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[83] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 51:1413–1541, 2004.

[84] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based construction of wavelet frames. *Applied and Computational Harmonic Analysis*, 14:1–46, 2003.

[85] C. Davis. All convex invariant functions of Hermitian matrices. *Archiv der Mathematik*, 8(4):276–278, 1957.

[86] P. Davis. *Circulant Matrices*. John Wiley and Sons, New York, 1979.

[87] Z. Denkowski, S. Migórski, and N. Papageorgiou. *An Introduction to Nonlinear Analysis: Applications*. Springer, Berlin, 2003.

[88] R. A. DeVore and B. J. Lucier. Fast wavelet techniques for near-optimal image processing. In *IEEE MILCOM '92 Conf. Rec.*, volume 3, pages 1129–1135, San Diego, CA, 1992. IEEE Press.

[89] R. A. DeVore and B. L. Lucier. Wavelets. *Acta Numerica*, 1:1–56, 1992.

[90] S. Didas, G. Steidl, and S. Setzer. Combined ℓ_2 data and gradient fitting in conjunction with ℓ_1 regularization. *Advances in Computational Mathematics*, 30(1), 2009.

[91] B. Dong and Z. Shen. Pseudo-splines, wavelets and framelets. *Applied and Computational Harmonic Analysis*, 22:78–104, 2007.

[92] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.

[93] R. J. Duffin and A. C. Schaeffer. A class of nonharmonic Fourier series. *Transactions of the American Mathematical Society*, 72:341–366, 1952.

[94] J. Eckstein. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, Massachusetts Institute of Technology, 1989.

[95] J. Eckstein. Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming. *Mathematics of Operations Research*, 18(1):202–226, 1993.

[96] J. Eckstein and D. P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.

[97] I. Ekeland and R. Temam. *Convex Analysis and Variational Problems*. North-Holland, Amsterdam, 1976.

[98] M. Elad and M. Aharon. Image denoising via sparse and redundant representation over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[99] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19:340–358, 2005.

[100] S. Esedoglu and S. Osher. Decomposition of images by anisotropic Rudin-Osher-Fatemi model. *Communications on Pure and Applied Mathematics*, 57:1609–1626, 2004.

[101] E. Esser. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. Technical report, UCLA Computational and Applied Mathematics, March 2009.

[102] E. Esser, X. Zhang, and T. F. Chan. A general framework for a class of first order primal-dual algorithms for TV minimization. Technical report, UCLA Computational and Applied Mathematics, August 2009.

[103] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume 2. Springer, New York, 2003.

[104] M. J. Fadili and J.-L. Starck. Sparse representations and Bayesian image inpainting. In *Signal Processing with Adaptive Sparse Structured Representations Workshop (SPARS'05)*, pages 417–424, Rennes, France, 2005.

[105] M. J. Farah. *The cognitive neuroscience of vision*. Wiley-Blackwell, Oxford, 2000.

[106] M. Figueiredo and J. Bioucas-Dias. Deconvolution of Poissonian images using variable splitting and augmented Lagrangian optimization. In *IEEE Workshop on Statistical Signal Processing*, Cardiff, 2009.

[107] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[108] K. Frick. *The Augmented Lagrangian Method and Associated Evolution Equations*. PhD thesis, 2008.

[109] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary–Value Problems*, volume 15 of *Studies in Mathematics and its Applications*, chapter 9, pages 299–331. North–Holland, Amsterdam, 1983.

[110] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2:17–40, 1976.

[111] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel. Towards PDE-based image compression. In B. M. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors, *Variational, Geometric, and Level Set Methods in Computer Vision*, volume 3752 of *Lecture Notes in Computer Science*, pages 37–48. Springer, Berlin, 2005.

[112] M. S. Gazzaniga, editor. *The cognitive neurosciences III*. MIT Press, Cambridge, MA, USA, 2004.

[113] G. Gilboa, J. Darbon, S. Osher, and T. F. Chan. Nonlocal convex functionals for image regularization. *UCLA CAM Report*, 06-57, 2006.

[114] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modelling & Simulation*, 7(3):1005–1028, 2008.

[115] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Birkhäuser, Basel, 1984.

[116] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, volume 9 of *SIAM Studies in Applied and Numerical Mathematics*. SIAM, Philadelphia, 1989.

[117] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975.

[118] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 2(27):622–645, 2005.

[119] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split Bregman method: Segmentation and surface reconstruction. Technical report, UCLA Computational and Applied Mathematics, February 2009.

[120] T. Goldstein and S. Osher. The Split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[121] N. I. M. Gould, D. Orban, and P. L. Toint. Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, pages 299–361, 2005.

[122] Y. Gousseau and J.-M. Morel. Are natural images of bounded variation? *SIAM Journal on Mathematical Analysis*, 33(3):634–648, 2001.

[123] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer, Dordrecht, 1995.

[124] Y. Gur and N. Sochen. Fast invariant Riemannian DT-MRI regularization. In *IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brasil, 2007.

[125] A. Haar. Zur Theorie der Orthogonalen Funktionen-Systeme. *Mathematische Annalen*, 69:331–371, 1910.

[126] E. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. *CAAM Technical Report TR07-07, Rice University*, 2007.

[127] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th ALVEY Vision Conference*, pages 147–152, Manchester, 1988.

[128] M. R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.

[129] W. Hintermüller and K. Kunisch. Total bounded variation regularization as a bilaterally constrained optimization problem. *SIAM Journal on Applied Mathematics*, 64(4):1311–1333, May 2004.

[130] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[131] V. I. Istrăţescu. *Fixed Point Theory*, volume 7 of *Mathematics and Its Applications*. D. Reidel Publishing Company, Dordrecht, 1981.

[132] A. N. Iusem. Augmented Lagrangian methods and proximal point methods for convex optimization. *Investigación Operativa*, 8:11–49, 1999.

[133] C. Kanzow, N. Yamashita, and M. Fukushima. Levenberg Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *Journal of Computational and Applied Mathematics*, 173:321–343, 2005.

[134] S. Kindermann, S. Osher, and P. W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modelling & Simulation*, 4(4):1091–1115, 2005.

[135] K. C. Kiwiel. Free-steering relaxation methods for problems with strictly convex costs and linear constraints. *Mathematics of Operations Research*, 22(2):326–349, 1997.

[136] K. C. Kiwiel. Proximal minimization methods with generalized Bregman functions. *SIAM Journal on Control and Optimization*, 35(4):1142–1168, 1997.

[137] C. Koch. *The Quest for Consciousness: A Neurobiological Approach*. Roberts & Company Publishers, Englewood, CO, USA, 2004.

[138] M. A. Krasnoselskii. Two observations about the method of successive approximations. *Uspekhi Matematicheskikh Nauk*, 10:123–127, 1955. In Russian.

[139] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[140] J. Lellmann, F. Becker, and C. Schnörr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[141] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In A. Lie, M. Lysaker, K. Morken, and X.-C. Tai, editors, *Second International Conference on Scale Space Methods and Variational Methods in Computer Vision, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, volume 5567 of *Lecture Notes in Computer Science*, pages 150–162. Springer, 2009.

[142] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.

[143] A. S. Lewis. The mathematics of eigenvalue optimization. *Mathematical Programming, Series B*, 97:155–176, 2003.

[144] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[145] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[146] I. Loris, M. Bertero, C. De Mol, R. Zanella, and L. Zanno. Accelerating gradient projection methods for ℓ_1-constrained signal recovery by steplength selection rules. *Applied and Computational Harmonic Analysis*, 27(2):247–254, 2009.

[147] L. Loss, G. Bebis, M. Nicolescu, and A. Skurikhin. An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds. *Computer Vision and Image Understanding*, 113:126–149, 2009.

[148] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. J. Wiley and Sons, Chichester, 1988.

[149] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, pages 1–8, June 2008.

[150] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems 21: 22nd Annual Conference on Neural Information Processing Systems 2008*. Curran Associates, Inc., Red Hook, NY, USA, December 2008.

[151] S. Mallat. Multiresolution approximation and wavelet orthonormal bases of $L_2(\mathbb{R}^d)$. *Transactions of the American Mathematical Society*, 315:69–87, 1989.

[152] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, second edition, 1999.

[153] W. R. Mann. Mean value methods in iteration. *Proceedings of the American Mathematical Society*, 16(4):506–510, 1953.

[154] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11:431–441, 1963.

[155] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Revue Française d'lnformatique et de Recherche Operationelle*, 4(3):154–158, 1970.

[156] G. Medioni, M. Lee, and C. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, Amsterdam, 2000.

[157] Y. Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, volume 22 of *University Lecture Series*. AMS, Providence, 2001.

[158] G. J. Minty. Monotone (nonlinear) operators in Hilbert space. *Duke Mathematical Journal*, 29:341–346, 1962.

[159] H. Mittelmann. An independent bechmarking of SDP and SOCP solvers. *Mathematical Programming Series B*, 95(2):407–430, 2003.

[160] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Societé Mathématique de France*, 93:273–299, 1965.

[161] P. Mrázek and J. Weickert. Rotationally invariant wavelet shrinkage. In B. Michaelis and G. Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 156–163, Berlin, 2003. Springer.

[162] A. Nemirovsky and D. Yudin. *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*. J. Wiley & Sons, New York, 1983.

[163] Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[164] Y. E. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer, Dordrecht, 2004.

[165] Y. E. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

[166] M. Nikolova. Image restoration by minimizing objective functions with nonsmooth data-fidelity terms. In *Proceedings of the first IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 11–18, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[167] M. Nikolova. Minimizers of cost functions involving nonsmooth data-fidelity terms. Applications to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2002.

[168] M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20:99–120, 2004.

[169] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

[170] Z. Opial. Weak convergence of a sequence of successive approximations for nonexpansive mappings. *Bulletin of the American Mathematical Society*, 73:591–597, 1967.

[171] G. B. Passty. Ergodic convergence to a zero of the sum of monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 72:383–390, 1979.

[172] D. W. Peaceman and H. H. Rachford. The numerial solution of parabolic elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):28–41, 1955.

[173] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

[174] S. Petra. *Semismooth least squares methods for complementarity problems*. PhD thesis, University of Würzburg, 2006.

[175] G. Plonka and G. Steidl. A multiscale wavelet-inspired scheme for nonlinear diffusion. *International Journal of Wavelets, Multiresolution and Information Processing*, 4(1):1–21, 2006.

[176] B. T. Polyak. *Introduction to Optimization*. Optimization Software, New York, 1987.

[177] B. Popilka, S. Setzer, and G. Steidl. Signal recovery from incomplete measurements in the presence of outliers. *Inverse Problems and Imaging*, 1(4):593–608, 2007.

[178] D. Potts and G. Steidl. Optimal trigonometric preconditioners for nonsymmetric Toeplitz systems. *Linear Algebra and its Applications*, 281:265–292, 1998.

[179] M. J. D. Powell. *Optimization*. Academic Press, New York, 1969. pages 283-298.

[180] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.

[181] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

[182] R. T. Rockafellar. The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and Applications*, 12:555–562, 1973.

[183] R. T. Rockafellar. *Conjugate Duality and Optimization*, volume 16 of *Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, PA, 1974.

[184] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.

[185] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*, volume 317 of *A Series of Comprehensive Studies in Mathematics*. Springer, Berlin, 2 edition, 2004.

[186] A. Ron and Z. Shen. Affine systems in $L_2(\mathbb{R}^d)$: The analysis of the analysis operator. *Journal of Functional Analysis*, 148:408–447, 1997.

[187] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[188] A. Sawatzky, C. Brune, F. Wübbeling, T. Kösters, K. Schäfers, and M. Burger. Accurate EM-TV algorithm in PET with low SNR. In *Nuclear Science Symposium Conference Record, 2008. NSS '08. IEEE*, pages 5133–5137. IEEE Press, October 2008.

[189] H. Schäfer. Über die Methode sukzessiver Approximationen. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 59:131–140, 1957.

[190] H. Scharr. *Optimal Operators in Digital Image Processing*. PhD thesis, University of Heidelberg, 2000.

[191] O. Scherzer and J. Weickert. Relations between regularization and diffusion filtering. *Journal of Mathematical Imaging and Vision*, 12:43–63, 2000.

[192] S. Setzer. Operator splittings, Bregman methods and frame shrinkage in image processing. *International Journal of Computer Vision*, 2009. accepted.

[193] S. Setzer. Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. In A. Lie, M. Lysaker, K. Morken, and X.-C. Tai, editors, *Second International Conference on Scale Space Methods and Variational Methods in Computer Vision, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, volume 5567 of *Lecture Notes in Computer Science*, pages 464–476. Springer, 2009.

[194] S. Setzer and G. Steidl. Variational methods with higher order derivatives in image processing. In M. Neamtu and L. L. Schumaker, editors, *Approximation XII, San Antonio, USA*, pages 360–386. Nashboro Press, Brentwood, 2008.

[195] S. Setzer, G. Steidl, B. Popilka, and B. Burgeth. Variational methods for denoising matrix fields. In D. H. Laidlaw and J. Weickert, editors, *Visualization and Processing of Tensor Fields, Advances and Perspectives*, Mathematics and Visualization, pages 341–360. Springer, Berlin, 2009.

[196] S. Setzer, G. Steidl, and T. Teuber. Restoration of images with rotated shapes. *Numerical Algorithms*, (48):49–66, 2008.

[197] S. Setzer, G. Steidl, and T. Teuber. Deblurring Poissonian images by split Bregman methods. *Journal of Visual Communication and Image Representation*, 2009. Accepted.

[198] S. Setzer, G. Steidl, T. Teuber, and G. Moerkotte. Approximation related to quotient functionals. *Journal of Approximation Theory*, 2009. Accepted for publication.

[199] J. Shi and S. Osher. A nonlinear inverse scale space method for a convex multiplicative noise model. *SIAM Journal on Imaging Sciences*, 1(3):294–321, 2008.

[200] G. Steidl. A note on the dual treatment of higher order regularization functionals. *Computing*, 76:135–148, 2006.

[201] G. Steidl, S. Didas, and J. Neumann. Splines in higher order TV regularization. *International Journal of Computer Vision*, 70(3):241–255, 2006.

[202] G. Steidl, S. Setzer, B. Popilka, and B. Burgeth. Restoration of matrix fields by second-order cone programming. *Computing*, 81:161 – 178, 2007.

[203] G. Steidl and T. Teuber. Anisotropic smoothing using double orientation. In A. Lie, M. Lysaker, K. Morken, and X.-C. Tai, editors, *Second International Conference on Scale Space Methods and Variational Methods in Computer Vision, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, volume 5567 of *Lecture Notes in Computer Science*, pages 477–489. Springer, 2009.

[204] G. Steidl and T. Teuber. Removing multiplicative noise by Douglas-Rachford splitting methods. *International Journal of Mathematical Imaging and Vision*, 2009. Accepted.

[205] G. Steidl, J. Weickert, T. Brox, P. Mrázek, and M. Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and SIDEs. *SIAM Journal on Numerical Analysis*, 42(2):686–713, 2004.

[206] X.-C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. In A. Lie, M. Lysaker, K. Morken, and X.-C. Tai, editors, *Second International Conference on Scale Space Methods and Variational Methods in Computer Vision, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, volume 5567 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2009.

[207] T. Teuber. Anisotropic diffusion and regularization methods for restoring rotated shapes. Diploma Thesis, University of Mannheim, 2008.

[208] T. Teuber, 2009. Private communication.

[209] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Preprint, National University of Singapore*, 2009.

[210] D. Tschumperlé and R. Deriche. Regularization of orthonormal vector sets using coupled PDE's. In *Proceedings of the first IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 3–10, Vancouver, Canada, July 2001. IEEE Computer Society Press.

[211] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29:119–138, 1991.

[212] P. Tseng and D. P. Bertsekas. Relaxation methods for problems with strictly convex costs and linear constraints. *Mathematics of Operations Research*, 16(3):462–481, 1991.

[213] R. S. Varga. *Matrix Iterative Analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer, New York, second edition, 2000.

[214] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, USA, 1995.

[215] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.

[216] J. von Neumann. Some matrix inequalities and metrization of matrix-space. In *Collected Works, Pergamon, Oxford, 1962, Volume IV, 205–218*, pages 286–300. Tomsk University Review, 1937.

[217] 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.

[218] Z. Wang, B. C. Vemuri, W. Chen, and T. H. Mareci. A constrained variational principle for direct estimation and smoothing of the diffusion tensor field from complex DWI. *IEEE Transactions on Medical Imaging*, 23(8):930–939, 2004.

[219] G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

[220] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

[221] J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pages 251–268. AMS, Providence, 2002.

[222] J. Weickert and H. Hagen, editors. *Visualization and Processing of Tensor Fields*. Springer, Berlin, 2006.

[223] P. Weiss, G. Aubert, and L. Blanc-Féraud. Efficient schemes for total variation minimization under constraints in image processing. *SIAM Journal on Scientific Computing*, 31(3):2047–2080, 2009.

[224] M. Welk, G. Steidl, and J. Weickert. Locally analytic schemes: A link between diffusion filtering and wavelet shrinkage. *Applied and Computational Harmonic Analysis*, 24:195–224, 2008.

[225] M. Welk and J. Weickert. Tensor field interpolation with PDEs. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 315–325. Springer, Berlin, 2006.

[226] M. Welk, J. Weickert, and G. Steidl. A four-pixel scheme for singular differential equations. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, Lecture Notes in Computer Science, pages 610–621. Springer, Berlin, 2005.

[227] H.-K. Xu. A variable Krasnosel'skii-Mann algorithm and the multiple-set split feasibility problem. *Inverse Problems*, 22:2021–2034, 2006.

[228] Q. Xu, A. W. Anderson, J. C. Gore, and Z. Ding. Diffusion tensor image smoothing using efficient and effective anisotropic filtering. In *IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brasil, 2007.

[229] Q. Yang and J. Zhao. Generalized KM theorems and their applications. *Inverse Problems*, 22:833–844, 2006.

[230] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.

[231] J. Yuan, C. Schnörr, and G. Steidl. Total-variation based piecewise affine regularization. In A. Lie, M. Lysaker, K. Morken, and X.-C. Tai, editors, *Second International Conference on Scale Space Methods and Variational Methods in Computer Vision, SSVM 2009, Voss, Norway, June 1-5, 2009. Proceedings*, volume 5567 of *Lecture Notes in Computer Science*, pages 552–564. Springer, 2009.

[232] M. Zéraï and M. Moakher. Riemannian curvature-driven flows for tensor-valued data. In F. Sgallari, A. Murli, and N. Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 592–602, Berlin, 2007. Springer.

[233] J. Zhao and Q. Yang. A note on the Krasnoselski-Mann theorem and its generalizations. *Inverse Problems*, 23:1011–1016, 2007.

[234] M. Zhu. *Fast numerical algorithms for total variation based image restoration*. PhD thesis, University of California, Los Angeles, USA, 2008.

[235] M. Zhu and T. F. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restauration. Technical report, UCLA Computational and Applied Mathematics, 2008.

[236] M. Zhu, S. J. Wright, and T. F. Chan. Duality-based algorithms for total variation image restoration. Technical report, UCLA Computational and Applied Mathematics, 2008.

Index

B-spline framelets, 131

algorithm
  alternating direction method of multipliers, 41
  augmented Lagrangian, 37
  backward-backward splitting, 26
  Bregman proximal point, 29
  classical proximal point, 34
  Douglas-Rachford splitting, 25
  FISTA, 71
  forward-backward splitting, 23
  MFISTA, 71
  Nesterov's, 70
  Peaceman-Rachford splitting, 27
  PIDSplit+, 85
  quadratic penalty, 37
alternating direction method of multipliers, *see* algorithm
anisotropy, 91
asymptotically regular, *see* operator
augmented Lagrangian algorithm, *see* algorithm
averaged, *see* operator

backward-backward splitting algorithm, *see* algorithm
Banach fixed point theorem, 14
Besov norm, 54
Besov space, 55
bounded variation, 55
Bregman distance, 29
Bregman function, 28
Bregman proximal point algorithm, *see* algorithm

classical proximal point algorithm, *see* algorithm
cocoercive, *see* operator
contractive, *see* operator
core, 11
coupled shrinkage, *see* shrinkage
coupled thresholding, *see* thresholding

diffusion tensor, 91
diffusion tensor magnetic resonance imaging, 111
Douglas-Rachford splitting algorithm, *see* algorithm
dual frame, 127

edge enhancing diffusion, 91

Fermat's rule, 11
firmly nonexpansive, *see* operator
FISTA, *see* algorithm
forward-backward splitting algorithm, *see* algorithm
frame, 127
framelets, 129
Frobenius norm, 113

I-divergence, 54
indicator function, 60
inpainting problem, 87
isometric, *see* operator

Jordan-product, 114

Karush-Kuhn-Tucker conditions, 11
Kullback-Leibler divergence, 54

Legendre conjugate, 9
level set, 32
Lipschitz continuous, *see* operator
locally semianalytic scheme, 92
lower semi-continuous, *see* operator

mask, 129
matrix-valued images, 111
maximal monotone, 12
MFISTA, *see* algorithm
monotone, 12
Moreau envelope, 8
multiresolution analysis, 129

Nesterov's algorithm, *see* algorithm
nonexpansive, *see* operator
nuclear norm, 117

oblique extension principle, 130
operator
    asymptotically regular, 16
    contractive, 14
    firmly nonexpansive, 15
    averaged, 14
    cocoercive, 24
    isometric, 14
    Lipschitz continuous, 14
    lower semi-continuous, 8
    nonexpansive, 14
    proximity, 8

Parseval frame, 127
Peaceman-Rachford splitting algorithm, *see* algorithm
PIDSplit+ algorithm, *see* algorithm
Poisson distribution, 53
positively homogeneous function, 60
proximity, *see* operator
proximum, 8

quadratic penalty algorithm, *see* algorithm
quasi-Féjer monotone sequence, 19

refinable function, 129
reflection operator, 25
relative interior, 11
resolvent, 13
Rudin-Osher-Fatemi model, 56

scaling function, 129
second-order cone programming, 120
shrinkage
    coupled, 59
    soft, 59
smoothness modulus, 54
Sobolev space, 56
soft shrinkage, *see* shrinkage
soft thresholding, *see* thresholding
subdifferential, 11
superresolution, 104
support function, 29
symmetric gauge function, 117

thresholding
    coupled, 59
    soft, 59
tight frame, 127
total variation semi-norm, 56
trace norm, 117

unitarily invariant matrix, 117
unitary extension principle, 130

wavelet function, 128

Yosida approximant, 27