Proximal methods
in
medical image reconstruction
and in
nonsmooth optimal control of
partial differential equations

Dissertation zur Erlangung
des naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Andreas Schindele

Eingereicht am: 28. April 2016
Betreuer: **Prof. Dr. Alfio Borzì**, Universität Würzburg

Erklärung/ Declaration

Hiermit erkläre ich, dass ich die eingereichte Doktorarbeit eigenständig, d.h. insbesondere selbständig und ohne Hilfe einer kommerziellen Promotionsberatung angefertigt und keine anderen als die von mir angegebenen Hilfsmitteln benutzt habe.

I hereby declare that I executed the thesis independently, i.e. in particular self-prepared and without the assistance of a commercial doctorate consultancy and that no sources and tools other than those mentioned have been used.

Ort/ Place, Datum/ Date   Unterschrift/ Signature

Zusammenfassung

Proximale Methoden sind iterative Optimierungsverfahren für Funktionale $J = J_1 + J_2$, die aus einem differenzierbaren Teil J_2 und einem möglicherweise nichtdifferenzierbaren Teil bestehen. In dieser Arbeit werden proximale Methoden für endlich- und unendlichdimensionale Optimierungsprobleme diskutiert. In endlichen Dimensionen lösen diese ℓ_1- und TV-Minimierungsprobleme welche erfolgreich in der Bildrekonstruktion der Magnetresonanztomographie (MRT) angewendet wurden. Die Konvergenz dieser Methoden wurde in diesem Zusammenhang bewiesen. Die vorgestellten proximalen Methoden wurden mit einer geteilten proximalen Methode verglichen und konnten ein besseres Signal-Rausch-Verhältnis erzielen. Zusätzlich wurde eine Anwendung präsentiert, die parallele Bildgebung verwendet.
Diese Methoden werden auch für unendlichdimensionale Probleme zur Lösung von nichtglatten linearen und bilinearen elliptischen und parabolischen optimalen Steuerungsproblemen diskutiert. Insbesondere wird die schnelle Konvergenz dieser Methoden bewiesen. Außerdem werden abgeschnittene proximale Methoden mit einem inexakten halbglatten Newtonverfahren verglichen. Die numerischen Ergebnisse demonstrieren die Effektivität der proximalen Methoden, welche im Vergleich zu den halbglatten Newtonverfahren in den meisten Fällen weniger Rechenzeit benötigen. Zusätzlich werden die theoretischen Abschätzungen bestätigt.

Abstract

Proximal methods are iterative optimization techniques for functionals, $J = J_1 + J_2$, consisting of a differentiable part J_2 and a possibly nondifferentiable part J_1. In this thesis proximal methods for finite- and infinite-dimensional optimization problems are discussed. In finite dimensions, they solve l_1- and TV-minimization problems that are effectively applied to image reconstruction in magnetic resonance imaging (MRI). Convergence of these methods in this setting is proved. The proposed proximal scheme is compared to a split proximal scheme and it achieves a better signal-to-noise ratio. In addition, an application that uses parallel imaging is presented.
In infinite dimensions, these methods are discussed to solve nonsmooth linear and bilinear elliptic and parabolic optimal control problems. In particular, fast convergence of these methods is proved. Furthermore, for benchmarking purposes, truncated proximal schemes are compared to an inexact semismooth Newton method. Results of numerical experiments are presented to demonstrate the computational effectiveness of our proximal schemes that need less computation time than the semismooth Newton method in most cases. Results of numerical experiments are presented that successfully validate the theoretical estimates.

Danksagung

Zuerst danke ich meinem Doktorvater Prof. Dr. Alfio Borzì dafür, dass er mir die Möglichkeit für diese Promotionsarbeit eröffnet hat. Er gab mir brilliante Ideen, sorgte für eine sehr angenehme und herzliche Arbeitsatmosphäre und hat mir mit seiner Erfahrung, seinem Verständnis und seiner Geduld den Weg gezeigt, der zu vier wissenschaftlichen Publikationen und dieser Dissertation geführt hat.

Des weiteren danke ich Prof. Dr. Herbert Köstler für die herzliche Zusammenarbeit in dem Projekt *Parallel Multigrid Imaging and Compressed Sensing for Dynamic 3D Magnetic Resonance Imaging* und den inspirierenden interdisziplinären Austausch.

Ein großer Dank geht an Dr. Tobias Wech und an Valentin Ratz, die mir als Schnittstelle zwischen der Mathematik und der Medizin eine große Hilfe waren. Durch diese Zusammenarbeit sind zwei medizinische Veröffentlichungen entstanden.

Ich bedanke mich auch bei allen Mitarbeitern des Lehrstuhls *Wissenschafliches Rechnen* der Julius-Maximilians-Universität Würzburg für eine herzliche Arbeitsatmosphäre und einem sehr unterstützenden Teamgeist. Besonders hervorheben möchte ich die ertragreichen Diskussionen mit Dr. Gabriele Ciaramella über die Anwendung Proximaler Methoden in der Optimalen Steuerung und die inspirierenden Gespräche mit meinem Bürokollegen Duncan Gathungu.

Besonders bedanken möchte ich mich bei meiner Familie, allen voran meinen Eltern Annemarie Schindele und Edmund Schindele, die mit ihrer Weisheit, ihrer Geduld und ihrer Anerkennung eine große Unterstützung für mich sind.

Ein unvergleichbarer Dank geht an meine Frau Sonja Eder. Ihre Liebe und ihre Präsenz sind wichtige Pfeiler meines Lebens, tragen mich durch herausfordernde sowie glückliche Zeiten und waren für das Entstehen dieser Arbeit unentbehrlich.

Zuletzt bedanke ich mich bei dem *Interdisziplinären Zentrum für klinische Forschung (IZKF)*, das mich während meiner Zeit in Würzburg finanziell unterstützt hat.

Contents

1. Introduction

The rise of compressed sensing in the last decade has paved the way for new possibilities in the field of signal acquisition and reconstruction, where l^1-based optimization and sparsity have been exploited to successfully recover 'functions' from few samples; see, e.g., [CRT06b, DE03]. In particular, it was shown [CW05] that l^1-based inverse problems in signal recovery can be very efficiently solved by proximal iterative schemes pioneered by Rockafellar [Roc76] and Nesterov [Nes83]. Nowadays, these iterative schemes are the method of choice in magnetic resonance imaging for solving finite dimensional optimization problems of the following form

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|\Phi x\|_1,$$

where the rectangular measurement matrix A represents a blur operator [LDP07], x is the signal to reconstruct, b is the measurement vector and Φ represents some sparsification matrix.

We remark that the research and successful application of proximal schemes is attracting attention of many scientists and practitioners, which results in many new developments in this field. We refer to, e.g., [LBR15] for recent results and additional references.

One of the most famous representative of proximal methods is the fast iterative soft thresholding algorithm (FISTA) that was introduced by Beck et al. in [BT11]. In addition to ℓ_1-minimization, proximal methods are also used to minimize total variations (TV) functionals [BT09] or a combination of TV and ℓ_1 minimization [HZM10] of the following form

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|\Phi x\|_1 + \mu \|x\|_{TV}. \tag{1.0.1}$$

This formulation plays an important role in image reconstruction due to the 'smoothness' of natural images.

Recently, Ochs et al. [OCBP14] introduced a variant of proximal method called *inertial proximal iterative algorithm for nonconvex optimization* (iPiano) to solve nonconvex problems of the following structure

$$\min_{x \in \mathbb{R}^n} f(x) + \beta \|x\|_1,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable bounded from below and possibly nonconvex.

One of the purposes of our work is the efficient solution of the ℓ_1-TV-optimization problem (1.0.1) with a proximal method and thus to contribute to the field of image reconstruction problems. Therefore, we introduce new optimization variables such that equation (1.0.1) can be solved by the FISTA scheme. This new algorithm is called FISTA-TV. We apply this method to medical MRI images.

Simultaneously to the development of ℓ_1-based optimization in finite dimensions, a great research effort has been made to solve infinite dimensional optimization problems governed by partial differential equations (PDEs); see, e.g., [BS11, Trö09, Ulb11] and references therein. In many cases, this research has focused on objective functionals with differentiable L^2 terms and non-smoothness resulted from the presence of control and state constraints. However, more recently, the investigation of L^1 cost functionals has become a central topic in PDE-based optimization [Sta09, WW10, IK03, CHW12], because they give rise to sparse controls that are advantageous in many applications like, e.g., optimal actuator placement [Sta09]. A representative formulation of optimal control problems with L^1 control costs is the following

$$\begin{aligned} \min_{(y,u)\in H_0^1(\Omega)\times L^2(\Omega)} \quad & \frac{1}{2}\|y-z\|_{L^2}^2 + \frac{\alpha}{2}\|u\|_{L^2}^2 + \beta\|u\|_{L^1} \\ \text{s.t.} \quad & c(y,u) = 0, \quad u_a \le u \le u_b \text{ a.e. in } \Omega, \end{aligned} \tag{1.0.2}$$

where $c(y,u) = 0$ represents a PDE for the state y including the control u. This problem has been discussed in, e.g., [Sta09, WW10, IK03, IK04] for the case where $c(y,u)$ represents a linear elliptic operator, in [CHW12] for the case where $c(y,u)$ represents a nonlinear elliptic operator, and in [HSW12, CCK13] for the case where $c(u,y)$ represents a parabolic operator. However, most of these works focus on PDEs with a linear control mechanism. An investigation of L^1 bilinear control problems in quantum mechanics can be found in [CB16]. Concerning the optimization methodology for (1.0.2), the semismooth Newton (SSN) method has been the solver of choice in all these references; see also the equivalent primal-dual active set method discussed in [IK04].

One of the purposes of our work is to combine the finite dimensional point of view with the infinite dimensional point of view and thus contribute to the field of PDE-based optimization with L^1 control costs by investigating proximal methods in the infinite-dimensional setting. In particular, we aim at implementing and analyzing proximal schemes for solving (1.0.2), where $c(y,u)$ is an elliptic or parabolic PDE with linear or bilinear control mechanism. Notice, that the latter has been a much less investigated problem. Our investigation is motivated by the fact that proximal methods may have a computational performance that is comparable to that of SSN methods. However, in contrast to the latter, proximal schemes do not require the construction of second-order derivatives and the implementation of, e.g., a Krylov solver.

We present a detailed implementation of different proximal schemes for solving our PDE control problems that is similar to the spirit of iPiano. Further, we extend the theoretical investigation in [OCBP14] for unconstrained finite-dimensional optimization problems, to our infinite-dimensional setting. In particular, we prove that our proximal schemes provide minimizing sequences that converge strongly to a local minimizer. Furthermore, we prove an $\mathcal{O}(1/\sqrt{k})$ convergence rate of the so-called proximal residual (that is closely related to a generalized gradient), where k is the number of proximal iterations. This notion of convergence is used in ℓ^1 -based optimization and in some application fields [WSS+16]. In addition, in a particular case, one can even prove an $\mathcal{O}(1/k^2)$ convergence rate of the value of reduced cost functional.

We remark that the application of proximal schemes to large-scale PDE control problems requires the iterative solution of the underlying PDEs. Therefore we focus on two inexact variants of our proximal schemes, where the PDE problems are solved up to a given tolerance and prove their convergence. For these variants, we obtain the same rate of convergence as in the exact case for a specific truncation strategy.

To validate our proximal schemes, we benchmark them with the state-of-the-art SSN scheme. However, in the case of large scale problems also a truncated version of the SSN scheme is required. We refer to it as the inexact SSN (ISSN) scheme and we prove its convergence for a specific truncation strategy.

We remark that many arguments in our analysis are similar to those presented in the finite-dimensional case. However, some additional arguments are necessary in infinite dimensions, especially regarding the structure of our differential constraints and the discussion of our inexact proximal schemes. We refer to [LBR15] for further results concerning the formulation of proximal schemes for infinite-dimensional optimization problems from a different perspective.

This thesis is organized into two parts. The first part covers proximal methods in the finite dimensional setting of image reconstruction and compressed sensing, whereas the second part addresses proximal methods in the infinite dimensional setting of sparse optimal control problems. The first part is subdivided in the following three chapters.

In Chapter 2, we discuss the role of sparse vectors in image compression and image reconstruction. Furthermore an introduction to compressed sensing is given, which is a mathematical theory of exact reconstruction of undersampled signals.

In Chapter 3, we discuss proximal methods in finite dimensions. These methods solve ℓ_1-minimization problems that arise in compressed sensing. In particular, a special proximal method, the FISTA and a corresponding $\mathcal{O}(1/k^2)$ convergence theorem is presented. We extend FISTA to FISTA-TV, such that it can also be used for a combination of ℓ_1- and total variation minimization. A theorem of convergence of the FISTA-TV method is proven.

In Chapter 4, the application of proximal methods for the reconstruction of magnetic resonance images is discussed. First, the theory of magnetic resonance imaging (MRI) is introduced. Then, the FISTA-TV is applied on 2D and 3D images and compared with another proximal method called FCSA that was introduced in [HZM10]. Lastly, our FISTA-TV method is adapted to a 4D real-time reconstruction of videos from mouse heartbeats. The second part of the thesis is organized in the following five chapters.

In Chapter 5, we discuss linear and bilinear elliptic and parabolic optimal control problems, where for completeness, some conditions for the existence of a unique control-to-state operator and its properties are considered.

Chapter 6 is devoted to the formulation and analysis of L^1 nonsmooth optimal control problems governed by elliptic and parabolic equations with linear and bilinear control mechanisms. In particular, we study conditions for convexity and the characterization of the optimal control solution as the solution to optimality systems for the linear and bilinear control cases.

In Chapter 7, we present a general truncated inertial proximal method (TIP) and four special variants of it, namely the CTIP and VTIP method, that differ in the choice of the stepsize strategy, and the FTP and FTPB method, that represent an infinite dimensional extension of the FISTA method. Furthermore the convergence of the function values is proven together with the convergence rate of the proximal residual. For the FTP and FTPB method the convergence rate of the objective values is shown to be $\mathcal{O}(1/k^2)$.

In Chapter 8, an ISSN method in function spaces is presented as the state of the art method for comparison purposes. For completeness, the theory of this method is extended to the case of elliptic and parabolic bilinear control problems.

In Chapter 9, a numerical comparison of the FTP, FTPB, CTIP, VTIP and ISSN schemes is presented. The results of this comparison demonstrate the competitiveness for our proximal schemes. Furthermore, results of numerical experiments are reported to validate our theoretical estimates.

A chapter of conclusion completes this work.

The results presented in this thesis formed the basis of the following publications

- A. Schindele and A. Borzì. Proximal Methods for Elliptic Optimal Control Problems with Sparsity Cost Functional. Applied Mathematics, 2016.
- A. Schindele and A. Borzì. Proximal methods for parabolic optimal control problems with a sparsity promoting cost functional, submitted.
- T. Wech, N. Seiberlich, A. Schindele, V. Grau, L. Diffley, M. L. Gyngell, A. Borzì, H. Köstler, and J. E. Schneider. Development of real-time magnetic resonance imaging of mouse hearts at 9.4 Tesla – simulations and first application. IEEE Transactions on Medical Imaging, 35(3):912–920, 2016.
- V. Ratz, T. Wech, A. Schindele, A. Dierks, A. Sauer, J. Reibetanz, A Borzì, T. Bley, H. Köstler. Dynamic 3D MR-Defecography, submitted.

Part I.

Proximal methods for finite-dimensional optimization problems

2. Sparsity and compressed sensing

The Shannon sampling theorem [Sha49] states that the sampling rate has to be at least twice as high as the maximum frequency of a signal in order to guarantee exact reconstruction of the signal from the sampling. If the sampling rate is below this threshold, the signal is called to be undersampled. The theory of compressed sensing, however, can guarantee exact reconstruction also for undersampled signals under some conditions. In this section, the mathematical theory of compressed sensing is introduced. For a more detailed discussion, we refer to [FR14, CRT06a]. We first give an overview of the concept of sparsity and image compression that are essential to understand compressed sensing.

2.1. Sparsity and image compression

In this section the term sparsity is defined and we introduce the compression of a sparse signal. Let $x \in \mathbb{C}^N$ be a complex valued vector. Then, we define

$$\|x\|_0 := |\operatorname{supp}(x)|,$$

where $\operatorname{supp}(x) := \{j : x_j \neq 0\}$. It is common to call $\|x\|_0$ the ℓ_0-norm of x, even if it does not fulfill the requirements of a quasi-norm. Now, we are able to define the k-sparsity of a vector $x \in \mathbb{C}^N$.

Definition 2.1.1. ***(k-sparsity)*** *The vector x is called k-sparse if $\|x\|_0 \leq k$ for $k > 0$. The set of k-sparse vectors is denoted by*

$$\Sigma_k := \{x \in \mathbb{C}^N : \|x\|_0 \leq k\}.$$

Furthermore, we define the best k-term approximation error in ℓ_p as follows.

Definition 2.1.2. *The best k-term approximation error in ℓ_p is defined by*

$$\sigma_k(x)_p := \inf_{z \in \Sigma_k} \|x - z\|_p,$$

where $\|\cdot\|_p$ denotes the p-norm, $\|v\|_p = (\sum_i v_i^p)^{1/p}$.

The signal x is called compressible if for $k << N$ the best k-term approximation error $\sigma_k(x)_p$ is reasonably small. In order to obtain a compressed signal $x_{[k]}$ where $\|x - x_{[k]}\|_p = \sigma_k(x)_p$, we use the rearrangement $r(x) := (|x_{i_1}|, \ldots, |x_{i_N}|)^T$, where $|x_{i_j}| \geq |x_{i_{j+1}}|$, $j = 1, \ldots, N-1$. Then, we have the following

$$\sigma_k(x)_p = \left(\sum_{j=k+1}^{N} r_j(x)^p \right)^{\frac{1}{p}}.$$

We construct

$$(x_{[k]})_i = \begin{cases} x_i & \text{for } |x_i| \geq r_k(x) \\ 0 & \text{else} \end{cases} .$$

This sparse vector satisfies the following

$$x_{[k]} = \arg\min_{z\in\Sigma_k} \|x - z\|_p.$$

2.2. Exact reconstruction of undersampled signals – compressed sensing

In image compression, one acquires the whole signal and then compresses it, thus costly acquired information is given away. Now, we only consider $m << N$ linear, nonadaptive measurements. The goal is to exactly reconstruct the signal from these incomplete measurements under the assumption of a small $\sigma_k(x)_p$.

The acquisition of m linear measurements $b \in \mathbb{C}^m$ is equivalent to applying the measurement matrix $A \in \mathbb{C}^{m\times N}$ on the signal $x \in \mathbb{C}^N$

$$b = A \cdot x.$$

If $\sigma_k(x)_p$ is not small enough but there exists a basis $(\phi_1, \ldots, \phi_N) = \Phi^T \in \mathbb{C}^{N\times N}$ and a $x_\Phi \in \mathbb{C}^N$ where $x = \Phi^T \cdot x_\Phi$ such that $\sigma_k(x_\Phi)_p$ is small, we have

$$b = A\Phi^T \cdot x_\Phi. \tag{2.2.1}$$

This system is highly underdetermined, since $m << N$. However, we have additional information on x_Φ since we assume that it is nearly k-sparse, or in other words $\sigma_k(x_\Phi)_p$ is small. So in order to reconstruct the signal from the measurements, one can calculate the sparsest vector x_Φ that solves (2.2.1). This problem is represented in the following combinatorial minimization problem

$$\min_{x\in\mathbb{C}^N} \|\Phi x\|_0 \quad \text{s.t.} \quad Ax = b,$$

or equivalently

$$\min_{z\in\mathbb{C}^N} \|z\|_0 \quad \text{s.t.} \quad A\Phi^T z = b, \tag{2.2.2}$$

where the reconstructed signal is given by $x = \Phi^T z^*$. This problem is in general NP-hard; see, e.g., [FR14]. Therefore the following convex relaxation is considered

$$\min_{x\in\mathbb{C}^N} \|\Phi x\|_1 \quad \text{s.t.} \quad Ax = b. \tag{2.2.3}$$

or equivalently

$$\min_{z\in\mathbb{C}^N} \|z\|_1 \quad \text{s.t.} \quad A\Phi^T z = b. \tag{2.2.4}$$

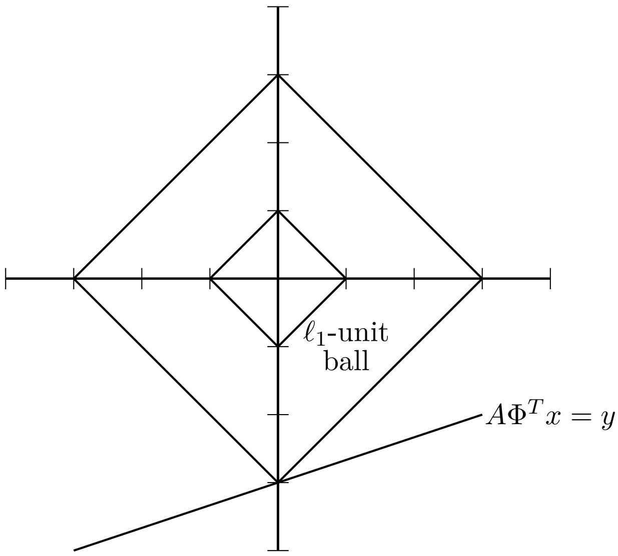


Figure 2.1.: Minimizing the ℓ_1-norm leads to sparsity.

As a motivation for this relaxation, one can consider the special case of N=2, m=1, that is illustrated in Figure 2.1.

Notice, that the solution of (2.2.2) coincides with the solution of (2.2.4) if the kernel of $A\Phi^T$ is not parallel to one of the surfaces of the ℓ_1 unit ball. This intuition of a connection between ℓ_1-minimization and sparsity will be analyzed in an exact way in the next section. From now on, we write $\hat{A} := A\Phi^T$ and refer to it as the measurement matrix.

2.2.1. Restricted isometry property

In this section, the connection between ℓ_1-minimization and sparsity, which is the essential idea of compressed sensing, is analyzed. The following property is needed.

Definition 2.2.1. *(Restricted isometry property – RIP)*
The restricted isometry constant δ_k of a matrix $\hat{A} \in \mathbb{C}^{m\times N}$ is the smallest number, such that the following holds

$$(1-\delta_k)\|z\|^2 \le \|\hat{A}z\|^2 \le (1+\delta_k)\|z\|^2, \tag{2.2.5}$$

for all $z \in \Sigma_k$, where we denote with $\|\cdot\| := \|\cdot\|_2$ for the 2-norm. The matrix $\hat{A}$ has the RIP of order k with constant δ_k if $\delta_k \in (0,1)$.

The following theorem states a connection between the ℓ_1-minimization problem with noisy measurements and the best k-term approximation error.

Theorem 2.2.1. *[Fou10, FR14] Let $\hat{A}$ fulfill the RIP of order k with constant*

$$\delta_{2k} < \frac{3}{4 + \sqrt{6}}$$

Furthermore, let $x \in \mathbb{C}^N$, $b = \hat{A}x + e$, $\|e\| \leq \eta$ and x^* the solution of

$$\min_{z \in \mathbb{C}^N} \|z\|_1 \quad s.t. \quad \|\hat{A}z - b\|^2 \leq \eta.$$

Then, we have

$$\|x - x^*\| \leq C_1 \eta + C_2 \frac{\sigma_k(x)_1}{\sqrt{k}},$$

where C_1 and C_2 only depend on δ_{2k}.

We see that if the product of measurement matrix and sparsification matrix $M\Phi$ fulfills the RIP, the ℓ_1-minimization provides a good signal reconstruction.

2.2.2. Coherence

In this section, the coherence is introduced, which is a helpful tool to analyze recovery ability of matrices in the special case of normed matrix columns.

Definition 2.2.2. *(Coherence) Let $\hat{A} = (a_1, \ldots, a_N) \in \mathbb{C}^{m \times N}$ be a matrix where $\|a_l\| = 1 \ \forall \ l \in \{1, \ldots, N\}$. Then, we define*

$$\mu := \max_{l \neq k} | \langle a_l, a_k \rangle |,$$

the coherence of $\hat{A}$.

Theorem 2.2.2. *[FR14] Let μ be the coherence of $\hat{A}$. Then, $\hat{A}$ fulfills the RIP of order k and $\delta_k \leq (k-1)\mu$.*

Several matrices with $\mu = \frac{1}{\sqrt{m}}$ are known, such as

$$\hat{A} = (I_m | F) \in \mathbb{C}^{m \times 2m}, \tag{2.2.6}$$

where I_m is the identity matrix and F is the unitary Fourier matrix $F_{ij} = \frac{1}{\sqrt{m}} \exp(2\pi(i-1)(j-1)k/m))$. From Theorem 2.2.2, we have that the restricted isometry constant of (2.2.6) is given by

$$\delta_k \leq \frac{k-1}{\sqrt{m}}, \tag{2.2.7}$$

and in order to use Theorem 2.2.1, the following must hold

$$\delta_{2k} \leq \frac{2k-1}{\sqrt{m}} < \delta, \tag{2.2.8}$$

which is equivalent to

$$(2k-1)^2 \leq \delta^2 \cdot m. \tag{2.2.9}$$

As far as we know, this quadratic dependence between k and m, could not be improved by now for deterministic matrices.

2.2.3. Random matrices

In order to improve the quadratic dependence between k and m, we introduce random matrices.

Definition 2.2.3. *(Random matrix) Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space and $X_{ij} : \Omega \to \mathbb{C}$, $i = 1, \dots, m$, $j = 1, \dots, N$ be random variables. Then, $\hat{A} := \hat{A}(\omega)$ where $\hat{A}_{ij} := X_{ij}(\omega)$ is called a random matrix.*

The following definition plays an important role to study the RIP of real-valued random matrices.

Definition 2.2.4 (Concentration inequality)**.** *Let $\hat{A} \in \mathbb{R}^{m\times N}$ be a random matrix where $\mathbb{E}(\|\hat{A}x\|^2) = \|x\|^2, \quad \forall x \in \mathbb{R}^N$ and $\mathbb{E}$ is the expectation value. Then, the concentration inequality for a constant $c_0 > 0$ is defined by*

$$\mathbb{P}\left(\left|\|\hat{A}x\|^2 - \|x\|^2\right| \geq \delta\|x\|^2\right) \leq 2e^{-c_0\delta^2 m}, \quad 0 < \delta < 1. \tag{2.2.10}$$

There are several random matrices that fulfill the concentration inequality, such as the Gauss matrix, whose entries $\hat{A}_{ij}$ are independent, identically distributed Gaussian random variables, $X_{ij} \sim N(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma^2 = 1/m$. Another example is the Bernoulli matrix, whose entries $\hat{A}_{ij}$ are independent, identically distributed $\pm k$-Bernoulli random variables with $k = 1/\sqrt{m}$. This means that each entry $\hat{A}_{ij}$ has the value $+1/\sqrt{m}$ or the value $-1/\sqrt{m}$ with the same probability.

The following theorem connects the concentration inequality with the restricted isometry property.

Theorem 2.2.3. *[FR14] Let $\hat{A} \in \mathbb{R}^{m\times N}$ be a random matrix that fulfills the concentration inequality (2.2.10) and let*

$$m \geq C\delta^{-2}(k\log(N/m) + log(\varepsilon^{-1})),$$

for some constant C, that only depends on c_0. Then, $\hat{A}$ fulfills the RIP (2.2.1) with the restricted isometry constant $\delta_k < \delta$ with a probability of $1 - \varepsilon$.

Consequently, the condition $\delta_{2k} < \delta$ from Theorem 2.2.1 holds with high probability, if

$$m \geq C' k \log(N/m),$$

with some constant $C' > 0$.

Another important class of random matrices are the random partial Fourier matrices, where m rows of a discrete Fourier matrix $F \in \mathbb{C}^{N\times N}$ with $F_{ij} = \frac{1}{\sqrt{N}} e^{2\pi(i-1)(j-1)k/N}$ are chosen randomly.

In applications, the random partial Fourier matrix plays an important role. Let therefore $x \in \mathbb{C}^N$ be a signal, that nearly consists of only k different frequencies, i.e. for $x_f = Fx$ the best k-term approximation error $\sigma_k(x_f)_1$ is small enough. We furthermore consider the measurement matrix $\hat{A}$ to be the random partial Fourier matrix. Then we have that the measurements $y \in \mathbb{C}^m$ are given by $y = \hat{A}x_f = Id^{m\times N} \cdot Fx_f = Id^{m\times N} \cdot x$, where $Id^{m\times N}$ are m randomly chosen rows of the identity matrix. So, the measurements y are m randomly chosen measurements of the frequencies of the original signal x.

The following theorem states how many measurements are needed for exact reconstruction.

Theorem 2.2.4. *[CRT06a] Let $\hat{A} \in \mathbb{C}^{m\times N}$ be a random partial Fourier matrix, $C \geq 29.6$ and let*

$$m \geq Ck\log(N/\varepsilon).$$

Then, the solution of the ℓ_1-minimization problem (2.2.4) is an exact reconstruction of an arbitrary vector $x \in \Sigma_k$ with probability greater or equal $1 - \varepsilon$.

3. Proximal methods

In Chapter 2, we have seen that the ℓ_1-minimization (2.2.3) of the form

$$\min_{x\in\mathbb{R}^{2N}} \|\Phi x\|_1 \quad \text{s.t.} \quad Ax = b, \tag{3.0.1}$$

often results in a sparse solution. The complex-valued space $\mathbb{C}^N$ is identified with the equivalent real-valued space $\mathbb{R}^{2N}$ and the corresponding equivalent l_p-norms. In the following, we will write $\mathbb{R}^N$ instead of $\mathbb{R}^{2N}$ for convenience.

Now, we consider the following unconstrained minimization problem

$$\min_{x\in\mathbb{R}^{N}} \lambda\|\Phi x\|_1 + \frac{1}{2}\|Ax - b\|^2. \tag{3.0.2}$$

In [DT06] it is shown that the solution x_λ of (3.0.2) is equal to zero if λ is big enough and that $\lim_{\lambda\to 0} x_\lambda = x^*$, where x^* is the solution of (3.0.1).

In this Chapter we discuss first-order proximal methods to solve a larger class of this nonsmooth optimization problem. Proximal methods originate from the proximal point algorithm introduced by Rockafellar in [Roc76], where the proximal function is used to solve a nonsmooth minimization problem of the general form

$$\min_{x\in\mathcal{H}} f(x),$$

where $f(x)$ is a lower semicontinuous, convex, and nondifferentiable functional and $\mathcal{H}$ is a Hilbert space. Nesterov [Nes07] as well as Beck and Teboule [BT11] developed two different methods that use an additional composite structure of the functional in order to accelerate the proximal point method.

3.1. Fast iterative soft thresholding algorithm – FISTA

In this section we focus on the proximal method of [BT11] that is called iterative soft thresholding algorithm (ISTA) and its fast extension FISTA.

The starting point to discuss proximal methods consists in identifying a smooth and a nonsmooth part in the objective functional. That is, we consider the following optimization problem

$$\min_{x\in\mathbb{R}^{N}} f_1(x) + f_2(x), \tag{3.1.1}$$

where $f_1(x)$ is continuous, convex, and possibly nonsmooth and $f_2(x)$ is a smooth, convex function with Lipschitz continuous gradient as follows

$$\|\nabla f_2(x) - \nabla f_2(y)\| \leq L(f_2)\|x-y\| \quad \forall x, y \in \mathbb{R}^N, \tag{3.1.2}$$

where $L(f_2) > 0$ is the Lipschitz constant. Notice, that our ℓ_1-minimization problem (3.0.2) has the additive structure (3.1.1), where $f_1(x) = \lambda\|\Phi x\|_1$ and $f_2(x) = \frac{1}{2}\|Ax - b\|_2$. The following lemma is essential in the formulation of proximal methods.

Lemma 3.1.1. *Let f_2 be differentiable, convex and it has Lipschitz continuous gradient with Lipschitz constant $L(f_2)$. Then, for all $L \geq L(f_2)$, we have*

$$f_2(x) \leq f_2(y) + \langle \nabla f_2(y), x-y\rangle + \frac{L}{2}\|x-y\|^2, \quad \forall x, y \in \mathbb{R}^N. \tag{3.1.3}$$

Proof.

$$\begin{aligned}
f_2(x) &= f_2(y) + \langle \nabla f_2(y), x-y\rangle + \int_0^1 \langle \nabla f_2(y + t(x-y)) - \nabla f_2(v), x-y\rangle \,\mathrm{dt} \\
&\leq f_2(y) + \langle \nabla f_2(v), x-y\rangle + \int_0^1 \|\nabla f_2(y+t(x-y)) - \nabla f_2(y))\| \,\|x-y\|\mathrm{dt} \\
&\leq f_2(y) + \langle \nabla f_2(v), x-y\rangle + \int_0^1 Lt\|x-y\|^2\mathrm{dt} \\
&\leq f_2(y) + \langle \nabla f_2(v), x-y\rangle + \frac{L}{2}\|x-y\|^2.
\end{aligned}$$

□

The following is valid for all $L \geq L(f_2) \in \mathbb{R}^+$. Furthermore if f_2 is twice differentiable, the Lipschitz constant of the gradient is given by $L(f_2) = \|\nabla^2 f_2\|_{l^2,l^2}$, see [RW97, Theorem 9.7]. This can be once evaluated by a power iteration [Wil88]. Because of Lemma 3.1.1, we have that

$$\min_{x\in\mathbb{R}^N} f_2(x) \leq \min_{x\in\mathbb{R}^N} \left\{ f_2(y) + \langle \nabla f_2(y), (x-y)\rangle + \frac{L}{2}\|x-y\|^2 \right\},$$

and

$$\arg\min_{x\in\mathbb{R}^N} \left\{ f_2(y) + \langle \nabla f_2(y), (x-y)\rangle + \frac{L}{2}\|x-y\|^2 \right\} = y - \frac{1}{L}\nabla f_2(y) =: s_{f_2}(y).$$

Therefore $1/L$ is the approximation to the optimal steplength for the steepest descent step to minimize f_2. Now, we can extend this method to the function $f = f_1 + f_2$ and obtain the following

$$\begin{aligned}
&\arg\min_{x\in\mathbb{R}^N} \left\{ f_1(x) + f_2(y) + \langle \nabla f_2(y), (x-y)\rangle + \frac{L}{2}\|x-y\|^2 \right\} \\
=&\arg\min_{x\in\mathbb{R}^N} \left\{ f_1(x) + \frac{L}{2}\left\|x - (s_{f_2}(y))\right\|^2 \right\} =: \mathrm{prox}_{f_1/L}(s_{f_2}(y)),
\end{aligned} \tag{3.1.4}$$

where we introduce the proximal function

$$\operatorname{prox}_f(y) = \operatorname*{arg\,min}_{x\in\mathbb{R}^N} \left\{ f(x) + \frac{1}{2}\|x-y\|^2 \right\}.$$

In general, it is impossible or too expensive to calculate the proximal function apart from particular f_1. In the particular case of (3.0.2), we have an explicit form of the proximal function as stated in Lemma 3.1.2. The soft thresholding function is defined in the following definition.

Definition 3.1.1. *We define the soft thresholding function by the following*

$$\mathbb{S}_\tau(y)_i := \begin{cases} y_i - \tau & \text{for } y_i > \tau \\ 0 & \text{for } |y_i| \le \tau \\ y_i + \tau & \text{for } y_i < -\tau \end{cases}.$$

Lemma 3.1.2. *Let $\Phi \in \mathbb{R}^{N\times N}$ be an orthogonal matrix, then the following holds*

$$\operatorname*{arg\,min}_{x\in\mathbb{R}^N} \left\{ \tau\|\Phi x\|_1 + \frac{1}{2}\|x-y\|^2 \right\} = \Phi^T \mathbb{S}_\tau(\Phi y) \quad \text{for any } \ y \in \mathbb{R}^N.$$

Proof. With the substitution $\hat{x} = \Phi x$ have that

$$\operatorname*{arg\,min}_x \{\tau\|\Phi x\|_1 + \frac{1}{2}\|x-y\|^2\} = \Phi^T \operatorname*{arg\,min}_{\hat{x}} \{\tau\|\hat{x}\|_1 + \frac{1}{2}\|\Phi^T\hat{x} - y\|^2\}.$$

Then, there exists a $\gamma(\hat{x}) \in \partial\|\hat{x}\|_1$, the subdifferential of $\|\cdot\|_1$, such that the solution $\hat{x} := \operatorname{arg\,min}_x \left\{ \tau\|x\|_1 + \frac{1}{2}\|\Phi^T x - y\|^2 \right\}$ fulfills the following variational inequality; see, e.g., [ET99];

$$\langle \hat{x} - \Phi y + \tau\gamma(\hat{x}), x - \hat{x} \rangle \ge 0, \quad \forall x \in \mathbb{R}^N. \tag{3.1.5}$$

Now, we show that $\hat{x} := \mathbb{S}_\tau(\Phi y)$ fulfills (3.1.5). The following investigation of the different cases is meant to be pointwise. We have

- $(\Phi y)_i - \tau > 0$:
  It follows that $\hat{x}_i = (\Phi y)_i - \tau > 0$ and $\gamma(\hat{x})_i = 1$ such that
  $(\hat{x}_i - (\Phi y)_i + \tau)(x_i - \hat{x}_i) = 0$.

- $|(\Phi y)_i| \le \tau$:
  It follows that $\hat{x}_i = 0$ and $\gamma(\hat{x})_i = \frac{(\Phi y)_i}{\tau} \in B_1(0)$ such that
  $\left(\hat{x}_i - (\Phi y)_i + \tau\left(\frac{(\Phi y)_i}{\tau}\right)\right)(x_i - \hat{x}_i) = 0$.

- $(\Phi y)_i + \tau < 0$:
  It follows that $\hat{x}_i = (\Phi y)_i + \tau < 0$ and $\gamma(\hat{x})_i = -1$ such that
  $(\hat{x}_i - (\Phi y)_i - \tau)(x_i - \hat{x}_i) = 0$. $\square$

Based on this lemma, we conclude that the solution to (3.1.4) is given by

$$\arg\min_{x\in\mathbb{R}^N}\left\{f_1(x)+\frac{L}{2}\left\|x-\left(y-\frac{1}{L}\nabla f_2(y)\right)\right\|^2\right\}=\Phi^T\mathbb{S}_{\frac{\lambda}{L}}\left(\Phi\left(y-\frac{1}{L}\nabla f_2(y)\right)\right),$$

that provides an approximation to the optimal x sought. Therefore we can use this result to define an iterative scheme as follows

$$x_k\leftarrow\Phi^T\mathbb{S}_{\frac{\lambda}{L}}\left(\Phi\left(x_{k-1}-\frac{1}{L}\nabla f_2(x_{k-1})\right)\right),$$

starting from a given x_0. The Algorithm 1 implements this proximal scheme, the so-called ISTA method.

This scheme is discussed in [BT11] and we give the following convergence result.

Theorem 3.1.3. *[BT11] Let $\{x_k\}$ be a sequence generated by Algorithm 1 and x^* be the solution to (3.0.2). Then, for every $k\geq 1$, the following holds*

$$f(x_k)-f(x^*)\leq\frac{L(f_2)\|x_0-x^*\|^2}{2k}.$$

Algorithm 1 (ISTA)

Require: λ, f_2, x_0, K
  Calculate $L=L(f_2)$;
  while $1\leq k\leq K$ **do**
    $x_k\leftarrow\Phi^T\mathbb{S}_{\frac{\lambda}{L}}\left(\Phi\left(x_{k-1}-\frac{1}{L}\nabla f_2(x_{k-1})\right)\right)$
  end while

Algorithm 2 (FISTA)

Require: λ, f_2, x_0, K
  Calculate $L=L(f_2)$; $y_0=x_0$; $t_0=1$
  while $1\leq k\leq K$ **do**
    $x_k\leftarrow\Phi^T\mathbb{S}_{\frac{\lambda}{L}}\left(\Phi\left(y_{k-1}-\frac{1}{L}\nabla f_2(y_{k-1})\right)\right)$
    $t_k=(1+\sqrt{1+4t_{k-1}^2})/2$
    $y_k=x_{k-1}+(\frac{t_{k-1}-1}{t_k})(x_{k-1}-x_k)$
  end while

In [Nes83], an acceleration strategy for proximal methods applied to convex optimization problems fulfilling (7.0.3) is formulated, that improves the rate of convergence of these schemes from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$. Specifically, one defines the sequence $\{t_k,y_k\}$ with

$$t_0=1,\qquad t_k:=1+\sqrt{1+4t_{k-1}^2}/2,\tag{3.1.6}$$

and

$$y_0:=x_0,\qquad y_k:=x_k+\frac{(t_{k-1}-1)}{t_k}(x_k-x_{k-1}).\tag{3.1.7}$$

Correspondingly, the optimization variable x_k is updated by the following

$$x_k\leftarrow\Phi^T\mathbb{S}_{\frac{\lambda}{L}}\left(\Phi\left(y_{k-1}-\frac{1}{L}\nabla f_2(y_{k-1})\right)\right).$$

This procedure FISTA is summarized in Algorithm 2.

We have the following error estimation for the FISTA-algorithm.

Theorem 3.1.4. *[BT11] Let $\{x_k\}$ be a sequence generated by Algorithm 2 and x^* be the solution of (3.1.1), then for every $k \geq 1$, the following holds*

$$f(x_k) - f(x^*) \leq \frac{2L(f_2)\|x_0 - x^*\|^2}{(k+1)^2}.$$

3.2. Total variations and l_1-minimization

In this section, the model function (3.0.2) is extended by a total variation term as follows

$$\min_{x \in \mathbb{R}^N} \lambda\|\Phi x\|_1 + \mu\|x\|_{TV_1} + \frac{1}{2}\|Ax - b\|^2, \tag{3.2.1}$$

where

$$\|x\|_{TV_1} = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \sum_{j=1}^{d} |\nabla_j \hat{x}_{i_1 \dots i_d}|,$$

with the finite differences $\nabla_j : \mathbb{R}^{n_1 \times \cdots \times n_d} \rightarrow: \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\nabla_j \hat{x}_{i_1 \dots i_d} := \hat{x}_{i_1 \dots i_j+1 \dots i_d} - \hat{x}_{i_1 \dots i_j \dots i_d}$ for $i_j \in \{1, \dots, n_j - 1\}$ and $\nabla_j x_{i_1 \dots i_d} := 0$ for $i_j = n_j$, $N = n_1 \cdot \cdots \cdot n_d$, $\lambda, \mu \geq 0$, and $\Phi \in \mathbb{R}^{N \times N}$ orthogonal. Here, we use the following bijective relation between $x \in \mathbb{R}^N$ and $\hat{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$

$$\hat{x}_{i_1, \dots, i_d} = x_{i_1 + \sum_{j=2}^{d}\left((i_j - 1)\prod_{k=1}^{j-1} n_k\right)}. \tag{3.2.2}$$

The optimization problem (3.2.1) results in good reconstructed images as shown in [LDP07] for a slightly different model function. In fact instead of the TV_1-seminorm, they use the isotropic TV_2-norm

$$\|x\|_{TV_2} := \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \sqrt{\sum_{j=1}^{d} \left(\nabla_j \hat{x}_{i_1 \dots i_d}\right)^2}.$$

Problem (3.2.1) was solved for $\lambda = 0$ in [BT09] by a dual method and in [OBG⁺05] by a split Bregman method. As far as we know, the most efficient algorithm to solve (3.2.1) is the fast composite splitting algorithm (FCSA) which was presented by Huang et al. in [HZM10]. The FCSA method is implemented in Algorithm 3.

The main difficulty of (3.2.1) is that there exists no explicit form of the proximal function of the combination of TV_1 and ℓ_1 minimization $f_1(x) = \lambda\|\Phi x\|_1 + \mu\|x\|_{TV_1}$. The FCSA calculates the proximal functions for the ℓ_1-norm and the TV_1-seminorm separately. The drawback of this method is the expensive estimation of the proximal function of the TV_1-seminorm.

In the algorithm that we present in this thesis, new optimization variables

$$\hat{g}^j \in \mathbb{R}^{n_1 \times \cdots \times n_d}, \quad \hat{g}^j_{i_1, \dots, i_d} := \nabla_j \hat{x}_{i_1, \dots, i_d}, \quad j \in \{1 \dots, d\}$$

are introduced in order to replace the TV_1-seminorm by the ℓ_1-norm. So the TV_1-seminorm becomes

$$\|x\|_{TV_1} = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \sum_{j=1}^{d} |\hat{g}^j_{i_1\dots i_d}| = \|(g^1,\dots,g^d)^T\|_1 =: \|g\|_1,$$

with $g \in \mathbb{R}^{N\cdot d}$, $g = (g^1,\dots,g^d)^T$ and the same equivalence between $\hat{g}^j$ and g^j as in (3.2.2).

With this setting, we arrive at the following optimization problem, which is equivalent to (3.2.1),

$$\min_{x,g} f_1(x,g) + f_2(x,g), \qquad \text{s.t.} \quad c(x,g) = 0, \quad l \le x \le u, \tag{3.2.3}$$

where

$$\begin{aligned} f_1(x,g) &:= \|(\lambda\Phi x, \mu g)^T\|_1, \\ f_2(x,g) &:= \frac{1}{2}\|Ax - b\|_2^2, \end{aligned}$$

and

$$c(x,g) := (\hat{\nabla}_1 x - g^1, \dots, \hat{\nabla}_d x - g^d)^T = \hat{\nabla} x - g,$$

where $\hat{\nabla}_j \in \mathbb{R}^N \to \mathbb{R}^N$ is obtained by the equivalence between $\hat{\nabla}_j x$ and $\nabla_j \hat{x}$, see (3.2.2). Furthermore, we define $\hat{\nabla} : \mathbb{R}^{N\cdot d} \to \mathbb{R}^{N\cdot d}$, $\hat{\nabla} x := (\hat{\nabla}_1 x, \dots, \hat{\nabla}_d x)^T$. We can solve this constrained optimization problem by using the differentiable penalty function $p(x,g) := \frac{1}{2}\|c(x,g)\|_2^2$ that results in the following optimization problem

$$\min_{x,g} f_1(x,g) + f_2(x,g) + \frac{\alpha}{2}\|c(x,g)\|_2^2, \qquad \text{s.t.} \quad l \le x_i \le u \tag{3.2.4}$$

where

$$\begin{aligned} f_1(x,g) &:= \|(\lambda\Phi x, \mu g)\|_1, \\ p_2(x,g) &:= \frac{1}{2}\|Ax - b\|_2^2 + \frac{\alpha}{2}\|c(x,g)\|_2^2, \quad \alpha \in \mathbb{R}^+ \end{aligned}$$

In order to obtain a relationship between (3.2.3) and (3.2.4) we use Theorem 5.6(e) in [CG02] to state the following.

Theorem 3.2.1. *The sequence (α_k) is strictly monotone increasing, $\alpha_k \to \infty$, the set $\{(x,g) \in \mathbb{R}^{(d+1)N},\ c(x,g) = 0\}$ is nonempty and (x_k, g_k) is a sequence of solutions of the optimization problem (3.2.4) with $\alpha = \alpha_k$. Then, the sequence (x_k, g_k) is converging to the unique solution of (3.2.3) and therefore x_k is converging to a unique solution of (3.2.1).*

Proof. According to Theorem 5.5(e) in [CG02], every accumulation point (x^*, g^*) of (x_k, g_k) is a solution of (3.2.3). Since $f_1 + p_2$ is strictly convex, it has a unique solution and therefore, every accumulation point of the sequence $(\bar{x}_k, \bar{g}_k)_{k\to\infty}$ is the unique solution of (3.2.4). Hence, $(\bar{x}_k, \bar{g}_k)_{k\to\infty}$ is converging to the solution (x^*, g^*) of (3.2.4). □

Now the total variation term can be written as a $\ell_1 - norm$ and thus it is possibly to apply Algorithm 2. Therefore, we separate the functions f_1 and s_{p_2} according to the variables x and g to obtain

$$f_1^x := \lambda \|\Phi x\|_1, \quad f_1^g := \mu \|g\|_1,$$

and

$$\begin{aligned} s_{p_2}^x(x,g) &:= x - \frac{1}{L} \nabla_x p_2(x,g) \\ s_{p_2}^g(x,g) &:= g - \frac{1}{L} \nabla_g p_2(x,g). \end{aligned}$$

Furthermore, we need the Lipschitz constant of the gradient of p_2 w.r.t. (x, g) given by

$$L(p_2) = \|\nabla^2 p_2\|_{l^2,l^2} = \left\| \begin{pmatrix} A^T A + \alpha \cdot \hat{\nabla}^T \hat{\nabla} & -\alpha \hat{\nabla}^T \\ -\alpha \hat{\nabla} & \alpha \cdot I_{N \cdot d} \end{pmatrix} \right\|_{l^2,l^2}.$$

These considerations are summarized in Algorithm 4 that implements our new FISTA-TV method.

Algorithm 3 (FCSA)

Require: A, b, x_0, λ, μ, l, u
  Calculate $L = \|A^T A\|_{l^2,l^2}$
  Set $y_0 = x_0$; $t_0 = 1$
  while $0 \le k \le K-1$ **do**
    $x_{k+1}^1 = \Phi^T \mathbb{S}_{\frac{\lambda}{L}} \left(\Phi \left(y_k - \frac{1}{L} \nabla_x p_2(y_k, h_k) \right) \right)$
    $x_{k+1}^2 = prox_{\frac{\mu \|x\|_{TV_1}}{L}} \left(y_k - \frac{1}{L} \nabla_x f_2(y_k, h_k) \right)$
    $x_{k+1} = (x_{k+1}^1 + x_{k+1}^2)/2$
    $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
    $y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}} (x_k - x_{k+1})$
  end while

Algorithm 4 (FISTA–TV)

Require: A, b, x_0, g_0, λ, μ, α, l, u
  Calculate $L = \|\nabla^2 p_2\|_{l^2,l^2}$
  Set $y_0 = x_0$; $h_0 = g_0$; $t_0 = 1$
  while $0 \le k \le K-1$ **do**
    $x_{k+1} = \Phi^T \mathbb{S}_{\frac{\lambda}{L}} \left(\Phi \left(y_k - \frac{1}{L} \nabla_x p_2(y_k, h_k) \right) \right)$
    $g_{k+1} = \mathbb{S}_{\frac{\mu}{L}} (h_k - \frac{1}{L} \nabla_g p_2(y_k, h_k))$
    $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
    $y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}} (x_k - x_{k+1})$
    $h_{k+1} = g_k + \frac{t_k - 1}{t_{k+1}} (g_k - g_{k+1})$
  end while

In the following theorem, the convergence of the FISTA-TV scheme is proved.

Theorem 3.2.2. *For every given accuracy $\varepsilon > 0$ there exists an $\alpha(\varepsilon)$ such that for every $\alpha > \alpha(\varepsilon)$ the FISTA-TV method provides a sequence x_k that converges to a limit $\bar{x}$ with $\|\bar{x} - x^*\| < \varepsilon$, where x^* is a minimizer of (3.2.1).*

Proof. Let α_k be a strictly monotone increasing sequence. According to Theorem 3.1.4, the FISTA-TV scheme provides a sequence that is converging to a minimizer $(\bar{x}_k, \bar{g}_k)$ of (3.2.4) for $\alpha = \alpha_k$. By using Theorem 3.2.1, we see that $\bar{x}_k$ is converging to the solution x^* of (3.2.1) as $k \to \infty$. Thus, for every $\varepsilon > 0$ there exists a K such that for all $k \ge K$ and therefore for all $\alpha_k \ge \alpha_K =: \alpha(\varepsilon)$ we have that $\|x^* - \bar{x}_k\| \le \varepsilon$. □

4. Proximal methods for image reconstruction – MRI

4.1. A Short introduction to MRI

In this section, we illustrate magnetic resonance imaging (MRI), which is a widely used imaging method to obtain clinical images of organs and soft tissues. For more detailed information, we refer to [WKM06].

MRI uses the properties of the hydrogen atom H^1. The nuclei of these atoms are protons and have an intrinsic spin with a magnetic moment showing in the direction of the spinning axis. If these nuclei are exposed to a strong magnetic field B_0 in the direction of the z-axis, their moments tend to align parallel to this field and add up to a measurable magnetization M_z. However, precession occurs, which means that the mean moments rotate around the z-axis with a specific frequency proportional to the strength of the magnetic field B_0. This frequency is called Larmor frequency $\omega_0 = \gamma B_0$, where $\gamma = 42.58$ MHz/T for the protons.

Now, assume the protons are in a stable state. By applying an electromagnetic wave of the same frequency as the Larmor frequency, the moments can be flipped by 90° into the x-y-plane, synchronously spinning around the z-axis. These transversal moments generate an alternating voltage of the same frequency as the Larmor frequency in a receiver coil, the magnetic resonance (MR) signal. The absence of phase differences between the so-called magnetic moments is called phase coherence. However, the MR signal reduces due to two independent processes T1 relaxation and T2 relaxation. T1 relaxation occurs because the transversal moments in the x-y-plane slowly realign with the magnetic field B_0 in the direction of the z-axis. T2 relaxation occurs because the phase coherence of the spinning transversal moments is lost after some time, and thus the nonsynchronous moments cancel each other.

In the different tissues of the body, the protons are part of different molecular structures such that the MR signals reduce at different speeds. This fact results in the contrast of the MR image.

Now, we know how the MR signal is produced and the remaining question is how to obtain an image from this signal. In particular, it is necessary to gain information about the spatial positions of the protons with the different MR signals. Therefore, the strength of the magnetic field B_0, and therefore the Larmor frequency of the protons, is not constant any more but varies in the z-direction. By choosing a specific frequency of the electromagnetic wave ω_{em}, one is exciting only the protons of a specific z-slice, where the strength of the magnetic field equals $B_z = \omega_{em}/\gamma$. In the y-direction, one is applying another variation of the strength of the magnetic field such that the precession frequency of the moments around the z-axis vary in the y-direction. After removing the variation in the y-direction, there is a phase difference between the rotation uniquely defining the y-position. Now, also in the remaining x-direction, one applies a magnetic variation such that the precession frequency uniquely defines the x-position. This phase-frequency space is called k-space. A 2D inverse Fourier transformation defined in Section 4.2, transforms the data from the fully-sampled k-space to the spatial space and provides the MR image. In the following section, we discuss undersampled data sets in the k-space.

4.2. Comparison of selected proximal methods in image reconstruction

In this section, we compare Algorithm 3 and Algorithm 4 to reconstruct undersampled 2D and 3D medical images. That means, instead of measuring the MR signal in the whole k-space, only some phases and frequencies are chosen according to a random mask. An example of this mask is shown in Figure 4.1. For the 2D comparison we are using the same images and parameters as in Huang et al. [HZM10]. We also use the algorithm FCSA published by Huang on his webpage[1]. The images of Figure 4.2 are also taken from this page.

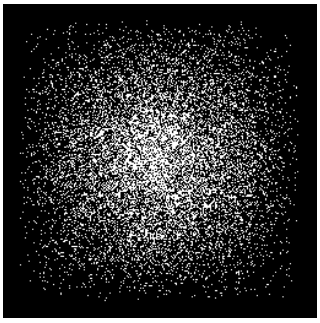


Figure 4.1.: Mask in the k-space.

4.2.1. 2D MRI reconstruction

We first apply the algorithm to the 2D images shown in Figure 4.2. We reconstruct an image with $N = 256 \cdot 256$ pixels with only $m << N$ measurements $b \in \mathbb{R}^m$. In our case we use the sampling rate $m/N \approx 0.158$. These measurements represent the m MR signals in the k-space that are chosen by some random mask shown in Figure 4.1. We use the same mask that is also used in [HZM10, MYZC08], consisting of randomly chosen information concentrated around the center in the k-space (low frequencies are more often chosen than high frequencies).

[1] `http://ranger.uta.edu/~huang/codes/FCSA_MRI1.0.rar`

The 2D Fourier matrix $F \in \mathbb{C}^{N\times N}$ is given by

$$(\hat{Fx})_{kl} := \frac{1}{\sqrt{N}} \sum_{m=0}^{n_1-1} \sum_{n=0}^{n_2-1} \hat{x}_{mn} \cdot \exp\left(-2\pi i \frac{km}{n_1}\right) \exp\left(-2\pi i \frac{nl}{n_2}\right),$$

where $\hat{x} \in \mathbb{C}^{n_1 \times n_2}$ is the equivalent 2 dimensional form of the signal $x \in \mathbb{C}^N$ as described in Section 3.2. Our measurement matrix is $A \in \mathbb{C}^{m\times N} = M \cdot F$ and the mask $M \in \{0,1\}^{m\times N}$ consists of m lines of the identity matrix corresponding to the white pixels in Figure 4.1. The matrix $\Phi \in \mathbb{C}^{N\times N}$ is chosen as the 2-dimensional wavelet transform, that lead to good sparsity for images as shown e.g. in [LDP07].

We measure the accuracy of the FCSA and FISTA-TV method by the signal-to-noise ratio defined by the following equation

$$\mathrm{SNR}(x, x_{\mathrm{ex}}) := 10 \log_{10} \frac{\mathrm{Var}(x_{\mathrm{ex}})}{\mathrm{E}[(x - x_{\mathrm{ex}})^2]}.$$

Here, the expectation value estimator is defined by $\mathrm{E}[x] := \frac{1}{N}\sum_{i=1}^{N} x_i$ and the variance estimator by $\mathrm{Var}(x) := \frac{1}{N-1}\sum_{i=1}^{N}(x_i - E[x])^2$ and x_{ex} is the exact image, x is the image we obtain from the algorithm.

We first compare the FCSA algorithm with the FISTA-TV where we additionally use the maximal range of possible greyscale values of the image $x \in [0, 255]$ by projecting x in the same way as Huang in [HZM10]. We have

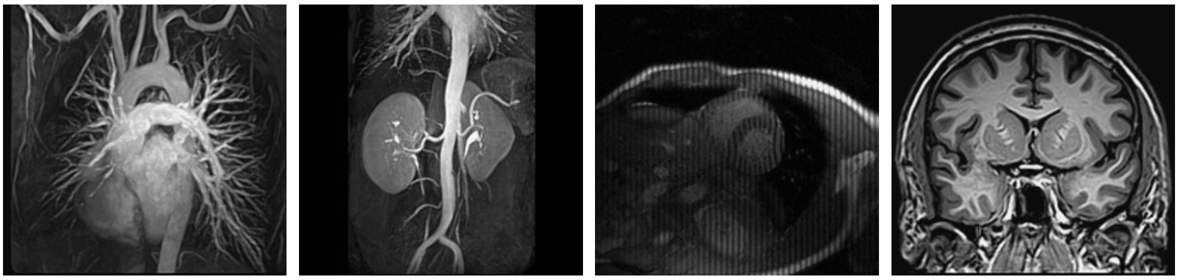
$$x \leftarrow \max\{\min\{x, 255\}, 0\}.$$

The chosen parameters are $\lambda = 0.035$, $\mu = 0.001$ as in [HZM10]. The starting value is always the zero vector $x_0 = (0, \dots, 0)$. We determine undersampled images from the exact images. The reconstructed images are shown in Figure 4.3 and Figure 4.4. The accuracy results and the convergence history are shown in Figure 4.5. We see that initially the FCSA iteration gains better results but after enough iterations the FISTA-TV method is much more accurate. The final signal-to-noise ratios are shown in Table 4.1.

4.2.2. 3D MRI reconstruction

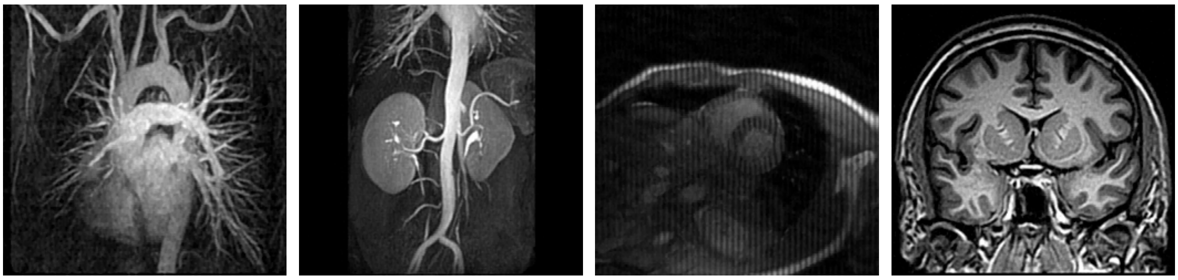
In this section, we compare the FCSA and FISTA-TV method for a 3D image reconstruction. We consider a three dimensional image of a human heart with $N = n_1 \cdot n_2 \cdot n_3 = 256 \cdot 256 \cdot 10$ pixels. So it consists of ten 2D image slices. Figure 4.6 shows the real part of four different slices of this image. $A \in \mathbb{C}^{m\times N}$ consists of m lines of the partial 3D Fourier transform and $\Phi \in \mathbb{C}^{N\times N}$ represents a 2-dimensional wavelet transformation for each of the ten slides as follows

$$\Phi := \begin{pmatrix} W & 0 & \dots & 0 \\ 0 & W & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & W \end{pmatrix},$$



(a) Chest (b) Renal arteries (c) Heart (d) Brain

Figure 4.2.: 2D Test Images

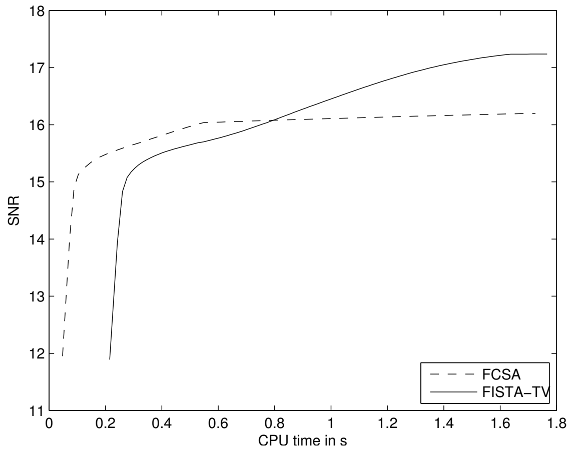


(a) Chest (b) Renal arteries (c) Heart (d) Brain

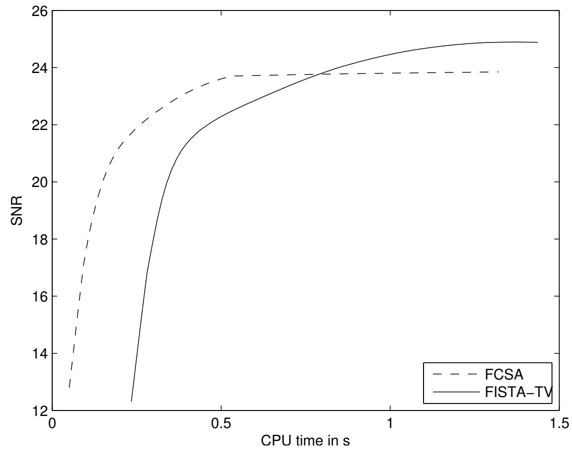
Figure 4.3.: 2D Reconstruction by the FCSA scheme

(a) Chest (b) Renal arteries (c) Heart (d) Brain

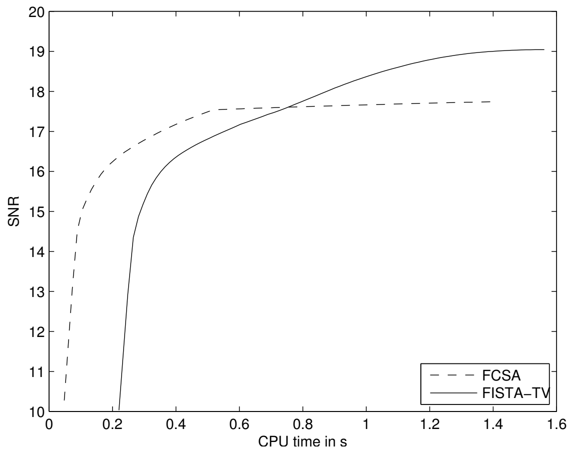
Figure 4.4.: 2D Reconstruction by the FISTA-TV scheme



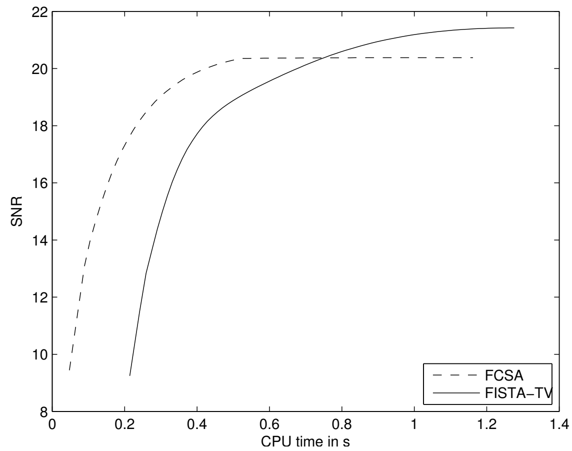
(a) Chest



(b) Renal arteries



(c) Heart



(d) Brain

Figure 4.5.: The signal-to-noise ratio of the FCSA and FISTA-TV algorithms for the 2D images.

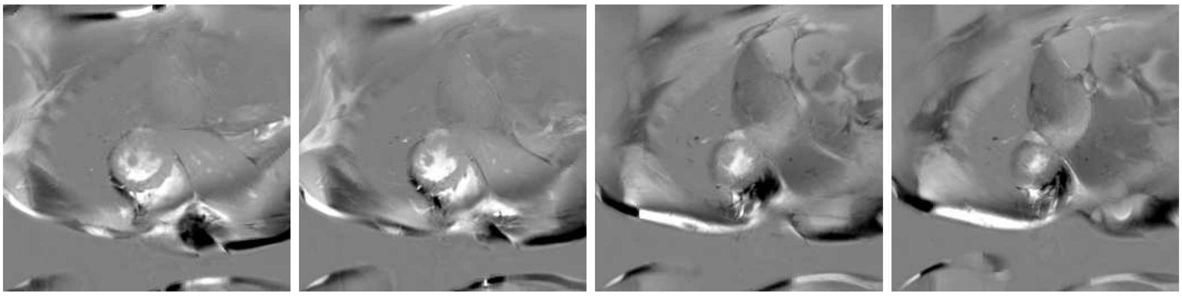


Figure 4.6.: 3D Test Images

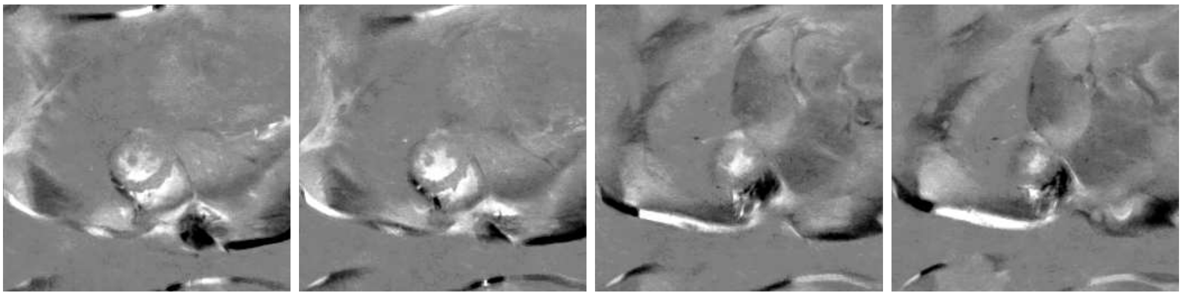


Figure 4.7.: 3D Reconstruction by the FCSA scheme

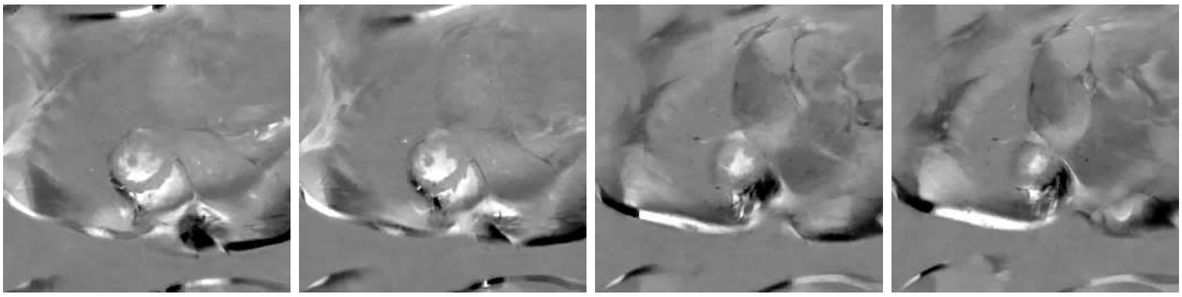


Figure 4.8.: 3D Reconstruction by the FISTA-TV scheme

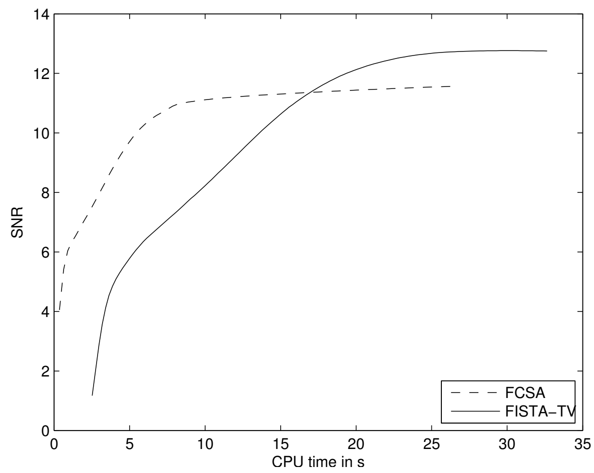


Figure 4.9.: The signal-to-noise ratio of the two algorithms for the 3D image

SNR	FCSA	FISTA-TV
Chest	16.2	17.2
Renal arteries	23.8	24.9
Heart	17.7	19.0
Brain	20.4	21.4
Heart 3D	18.35	24.48

Table 4.1.: Comparison of the SNR between FCSA and FISTA-TV schemes.

where $W \in \mathbb{C}^{(n_1 \cdot n_2) \times (n_1 \cdot n_2)}$ is the discrete two-dimensional wavelet transform. We use again a random mask which is concentrated around the center in the frequency domain. The sampling rate is $m/N \approx 0.225$. The chosen parameters are $\lambda = 0.008$, $\mu = 0.03$ and $\alpha = 0.3$. The reconstructed images are shown in Figure 4.7 and Figure 4.8 and the corresponding signal-to-noise ratios and the convergence history in Figure 4.9.

We observe that in the first 17s the FCSA method leads to better results than our FISTA-TV scheme. However, after 17s the FISTA-TV increases the SNR and the solution is more accurate than the best possible solution of FCSA.

4.3. A MR application with parallel imaging

Our FISTA-TV was successfully applied to assess real-time information of the cardiac function in mice from parallel coil data. The results are published in [WSS$^+$16]. In this work, several receiver coils are placed side by side for the simultaneous acquisition of the MR signal that consists of undersampled radial measurements, the projections. On this data the generalized radial autocalibrating partially parallel acquisitions (GRAPPA) technique [SED$^+$11] is applied in order to increase the number of initial projections per time frame. Then, the radial information is assigned from the radial grid to a Cartesian grid by GRAPPA operator gridding [SBB+08] which exploits the variation of the coil sensitivities to perform small changes in k-space. This results in the Cartesian undersampled multicoil k-space information b. In the linearly segmented (LS) case, the projections were equiangularly distributed with an increment between adjacent projections of $\Delta\vartheta = \pi/n_{proj}$, where n_{proj} is total number of projections. In addition, a Golden Angle (GA) acquisition is discussed, where the increment between consecutive projections was set to $\Delta\vartheta = 2\pi/(\sqrt{5}+1)$, guaranteeing optimal profile distribution for any arbitrary number of projections used in the reconstruction.

We consider the following minimization problem to determine fully sampled data.

$$\min_x \|\nabla_t Sx\|_1 + \frac{\mu}{2}\|b - Ax\|_2^2, \tag{4.3.1}$$

where $x \in \mathbb{C}^{N\times n_{coils}}$ is the resulting fully sampled image from the n_{coils} different coils, $S : \mathbb{C}^{N\times n_{coils}} \to \mathbb{C}^N$ holds the information of the coil sensitivities and is a coil combination operation [WMG00] resulting in one single complex valued image, A is a discrete Fourier transform for each coil and each time frame and b is the undersampled k-space information after applying GRAPPA and GRAPPA operation gridding. Furthermore ∇_t denotes the temporal, discrete total variation operator.

To solve (4.3.1) we use the strategy developed in Chapter 3 and apply the FISTA-TV algorithm. Therefore, we introduce new optimization variables g such that with

$$\begin{aligned} c(x,g) &= \nabla_t Sx - g, \\ p_2 &= \frac{\alpha}{2}\|c(x,g)\|_2^2 + \frac{\mu}{2}\|b - Ax\|_2^2. \end{aligned}$$

Further, we have

$$L(p_2) = \left\| \begin{pmatrix} \mu A^T A + \alpha \cdot S^T {\nabla_t}^T \nabla_t S & -\alpha S^T {\nabla_t}^T \\ -\alpha \nabla_t S & \alpha \cdot I_{N\cdot d} \end{pmatrix} \right\|_{l^2,l^2}.$$

With this setting, Algorithm 4 provides a solution to the following minimization problem

$$\min_{x,g} \|g\|_1 + \frac{\mu}{2}\|b - Ax\|_2^2 + \frac{\alpha}{2}\|c(x,g)\|_2^2, \tag{4.3.2}$$

whose solution, for sufficiently large α, is close to a solution of (4.3.1).

It is shown in [WSS+16] that this method enables real time cardiac imaging in mice, significantly reduces the scan time, and enables the investigation of small animal models with ventricular arrhythmias for the first time.

Representative end-diastolic (top row) and end-systolic (bottom row) frames are shown in Figure 4.10 for conventional (left column) and real-time acquisitions, respectively. While the left-ventricular wall and cavity were well resolved in all cases, the linear sampling schemes showed better image quality compared to the golden angle acquisitions.

The left-ventricular functional parameters obtained by blinded analysis in a mid-ventricular slice are listed in Table 4.2, and show generally good agreement between real-time undersampled and fully-sampled data. The spread of the left-ventricle mass as measured in the real-time data was larger than for the conventional data, while it was comparable for the volumes.

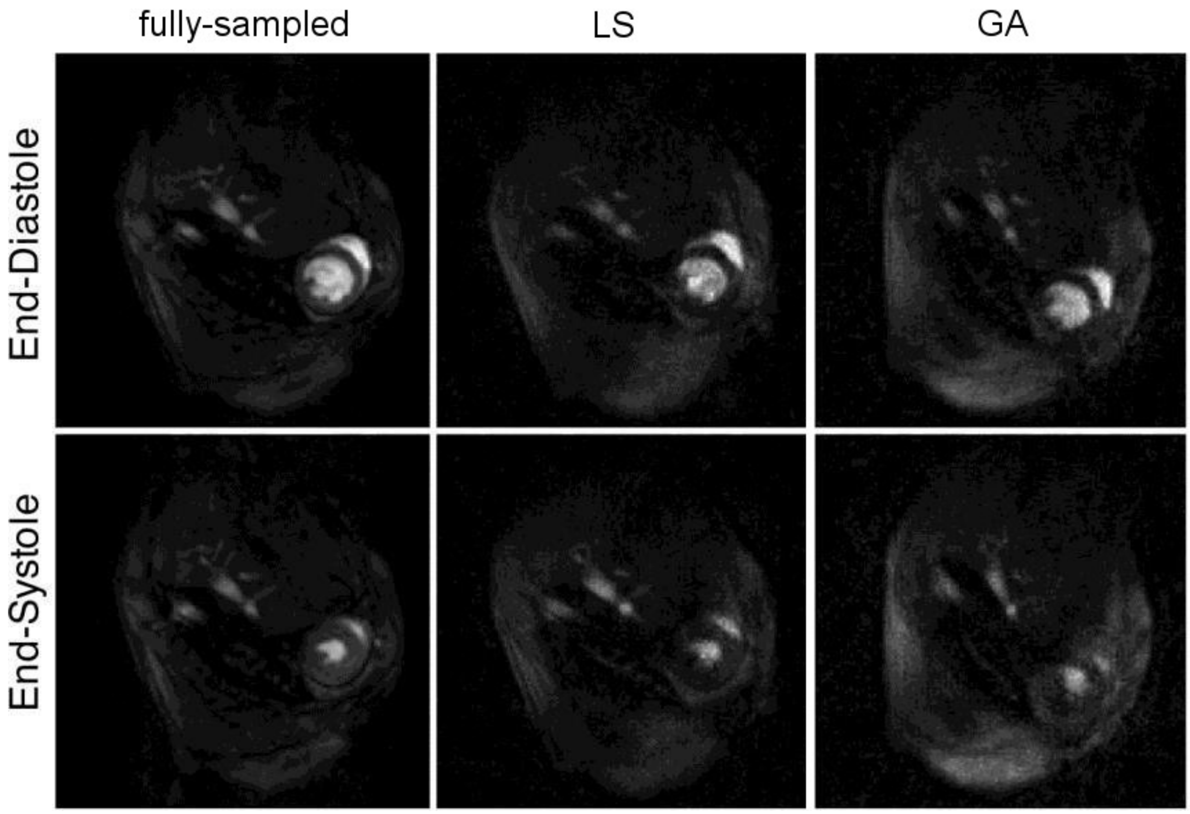


Figure 4.10.: Mid-ventricular slice through the same mouse thorax showing the heart in short-axis orientation, acquired with a prospectively-gated Cartesian multiframe sequence (left column) and with the radial real-time sequence. Top row: end-diastole; bottom row: end-systole. Scale bar - 5 mm.

	fully-sampled	LS	GA
end-diastolic mass in mg	14.5 ± 0.8	16.9 ± 2.5	17.9 ± 2.9
end-diastolic volume in μl	9.6 ± 1.2	9.1 ± 1.0	8.8 ± 2.2
end-systolic mass in mg	18.5 ± 0.5	20.1 ± 2.8	18.6 ± 2.9
end-systolic volume in μl	2.9 ± 1.5	2.5 ± 1.3	1.7 ± 0.8
stroke volume in μl	6.7 ± 1.1	6.6 ± 0.9	7.1 ± 2.6
ejection fraction in %	70 ± 13	73 ± 12	79 ± 12

Table 4.2.:

Part II.

Proximal methods for infinite-dimensional optimization problems

5. Partial differential equation models

In this section, we discuss elliptic and parabolic PDE models with linear and bilinear control structures. Notice that these models are already discussed in many references; see, e.g., [Eva10, KV09, Lio71, Trö09]. However, our focus is the presence of a control function that will be determined by proximal schemes.

5.1. Elliptic models

We start our discussion with linear elliptic equations.

5.1.1. Linear control mechanism

Consider the following boundary value problem

$$\begin{cases} Ay + u = f & \text{in } \Omega \\ \qquad y = 0 & \text{on } \partial\Omega, \end{cases} \tag{5.1.1}$$

where $\Omega \subset \mathbb{R}^n$, with $n \leq 3$, is a bounded domain and $f \in L^2(\Omega)$. The operator $A : H_0^1(\Omega) \to H^{-1}(\Omega)$ represents a second-order linear elliptic differential operator of the following form

$$Ay = -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_j}\left(a_{ij}\frac{\partial}{\partial x_j} y\right) + a_0 y,$$

such that $a_{i,j}, a_0 \in L^\infty(\Omega)$, and $a_{i,j}$ satisfies the coercivity condition

$$\sum_{i,j=1}^{n} a_{ij}(x)\xi_1\xi_2 \geq \theta \sum_{j=1}^{n} \xi_j^2 \quad \text{a.e. in } \Omega, \tag{5.1.2}$$

for some $\theta > 0$ and $a_0 \geq 0$, for any $\xi_1, \xi_2 \in \mathbb{R}$. For the existence and uniqueness of solutions to (5.1.1) see [Eva10, Section 6].

5.1.2. Bilinear control mechanism

Further, we consider the following bilinear elliptic control problem

$$\begin{cases} Ay + uy = f & \text{in } \Omega \\ \qquad y = 0 & \text{on } \partial\Omega. \end{cases} \tag{5.1.3}$$

In both linear and bilinear control settings, we require $u \in U_{ad}$, with the following set of feasible controls

$$U_{ad} := \{u \in L^2(\Omega) : u_a \leq u \leq u_b \text{ a.e. in } \Omega\}, \tag{5.1.4}$$

where $u_a \leq 0 \leq u_b$, $u_a < u_b$.

Now, we discuss the existence of a unique weak solution to (5.1.3). For this purpose, we need the Poincaré-Friedrichs lemma; see, e.g., [Eva10].

Lemma 5.1.1. *(Poincaré-Friedrichs inequality) Let $\Omega \subset \mathbb{R}^n$ be a bounded Lipschitz domain, which is contained in the cube $C := I_1 \times \cdots \times I_n$, where $I_1, \dots I_n$ are intervals and let $c_\Omega := \left(\sum_{i=1}^n \frac{2}{|I_i|^2}\right)^{-1}$, then*

$$\int_\Omega y^2 dx \leq c_\Omega \int_\Omega |\nabla y|^2 dx, \tag{5.1.5}$$

holds for all $y \in H_0^1(\Omega)$.

We denote with $\|\cdot\| := \|\cdot\|_{L^2(\Omega)}$ for the $L^2(\Omega_T)$-norm in the Hilbert space $L^2(\Omega)$ induced by the inner product $\langle \cdot, \cdot \rangle := \langle \cdot, \cdot \rangle_{L^2(\Omega)}$.

Theorem 5.1.2. *Let $u \in U_{ad}$, where*

$$u_a > -a_0 - \frac{\theta}{c_\Omega}. \tag{5.1.6}$$

Then, there exists a unique weak solution $y \in H_0^1(\Omega)$ to the bilinear elliptic problem (5.1.3) and the following property holds

$$\|y\|_{H^1(\Omega)} \leq C_1 \|f\|, \tag{5.1.7}$$

Proof. Problem (5.1.3) can be written as follows

$$\langle Ay + uy, v \rangle = \langle f, v \rangle \quad \text{for all } v \in H_0^1(\Omega).$$

We define $a(y, v) := \langle Ay + uy, v \rangle$ and $\kappa := \frac{u_a + a_0 + \theta / c_\Omega}{c_\Omega + 1} > 0$, then by using (5.1.2), $u \geq u_a$, and (5.1.5), we have

$$\begin{aligned}
a(y,y) = \langle Ay + uy, y \rangle &= \int_\Omega \left(\sum_{i,j=1}^n a_{ij} \frac{\partial y}{\partial x_j} \frac{\partial y}{\partial x_i} + (a_0 + u) y^2 \right) \mathrm{dx} \\
&\geq \int_\Omega \left(\theta |\nabla y|^2 + (a_0 + u_a) y^2 \right) \mathrm{dx} \\
&= \kappa c_\Omega \int_\Omega |\nabla y|^2 \mathrm{dx} + (\theta - \kappa c_\Omega) \int_\Omega |\nabla y|^2 \mathrm{dx} + \int_\Omega (a_0 + u_a) y^2 \mathrm{dx} \\
&\geq \kappa c_\Omega \int_\Omega |\nabla y|^2 \mathrm{dx} + \left(\frac{\theta}{c_\Omega} - \kappa \right) \int_\Omega y^2 \mathrm{dx} + \int_\Omega (a_0 + u_a) y^2 \mathrm{dx} \\
&\geq \min \left(\kappa c_\Omega, u_a + a_0 + \frac{\theta}{c_\Omega} - \kappa \right) \|y\|_{H^1}^2 = \kappa c_\Omega \|y\|_{H^1}^2.
\end{aligned}$$

In the forth line the Poincaré-Friedrichs was used. With $\kappa := \frac{u_a+a_0+\theta/c_\Omega}{c_\Omega+1} > 0$, we have that $\kappa c_\Omega = u_a + a_0 + \theta/c_\Omega - \kappa$ and thus, we have

$$\langle Ay + uy, y\rangle \geq C_0\|y\|_{H^1}^2 \text{ for } C_0 := \kappa c_\Omega, \tag{5.1.8}$$

and therefore

$$\|y\|^2 \leq \|y\|_{H^1}^2 \leq C_0^{-1}\langle Ay + uy, y\rangle = C_0^{-1}\langle f, y\rangle \leq C_0^{-1}\|f\|\|y\|.$$

Therefore we obtain (5.1.7) with $C_1 := C_0^{-1}$. Furthermore, one has

$$|a(y,v)| \leq \left(\sum_{i,j}\|a_{ij}\|_{L^\infty} + \|a_0 + u_b\|_{L^\infty}\right)\|y\|_{H^1}\|v\|_{H^1}.$$

We can use the Lemma of Lax-Milgram with $V = H_0^1(\Omega)$ to complete the proof. □

Remark 5.1.1. *In the case of $\Omega = (0,1)^n$, $n \leq 3$, and $A = -\Delta$, including homogeneous Dirichlet boundary conditions, we have a_0=0, $\theta = 1$ and $c_\Omega = 1/2n$ such that we can ensure invertibility for $u_a > -2n$.*

Remark 5.1.2. *In order to ensure a unique solution, we require condition (5.1.6) for the choice of u_a in the bilinear case.*

Next, we recall the following theorem stating higher regularity of solutions to (5.1.3); see [Gri85, Theorem 4.3.1.4].

Theorem 5.1.3. *Let $\Omega \subset \mathbb{R}^n, n \leq 3$, be a convex and bounded polygonal or polyhedral domain. If in addition to the assumptions of Theorem 5.1.2, we have that $a_{i,j} \in C^1(\bar{\Omega})$, then $y \in H^1(\Omega) \cap H^2(\Omega)$ and the following holds*

$$\|y\|_{H^2(\Omega)} \leq \tilde{C}\left(\|f\| + \|y\|\right) \leq C\|f\|, \tag{5.1.9}$$

for some appropriate constants $C, \tilde{C} > 0$ that only depend on Ω.

Remark 5.1.3. *Because $H^2(\Omega)$ can be embedded in $L^\infty(\Omega)$ [Ada75], for $n \leq 3$ and using (5.1.9), we obtain*

$$y \in L^\infty(\Omega) \quad \textit{and} \quad \|y\|_{L^\infty(\Omega)} \leq C\|f\|. \tag{5.1.10}$$

Theorem 5.1.2 and Theorem 5.1.3 ensure the existence of a unique control-to-state operator

$$S : U_{ad} \to H_0^1(\Omega) \cap H^2(\Omega), \quad u \mapsto S(u), \tag{5.1.11}$$

where in the linear case $S(u) = A^{-1}(f - u)$ represents the unique solution to (5.1.1) and in the bilinear case $S(u) = (A + u)^{-1}f$ is the unique solution to (5.1.3).

Remark 5.1.4. *For the bilinear case, the control-to-state operator $S(u)$ is not Fréchet-differentiable in the L^2 topology since for every $\varepsilon > 0$ there is always an $h \in L^2(\Omega)$ with $\|h\| \leq \varepsilon$ such that $u + h \notin U_{ad}$ and therefore it is not necessarily defined. However, we only need the following weaker form of differentiability, which is a directional differentiability in all $v \in U_{ad}$ in the directions $(u - v)$ for $u \in U_{ad}$. This is called Q-differentiability; see [KV09].*

Definition 5.1.1. ***(Q-Differentiability).*** *Let $U \subset X$ be a convex set and $T : U \to Y$. Then T is called to be Q-differentiable in $v \in U$, if there exists a mapping $T'_U(v) \in \mathcal{L}(X, Y)$, such that for all $u \in U$ the following holds*

$$\frac{\|T(v + u - v) - T(v) - T'_U(v)(u - v)\|_Y}{\|u - v\|_X} \to 0 \quad \text{if } \|u - v\|_X \to 0.$$

In the following, we omit the index U and write $T' = T'_U$.

The Q-derivatives of $S(u)$ have the following properties in the bilinear case.

Lemma 5.1.4. *The control-to-state-operator S is at least two times Q-differentiable in U_{ad} and its derivatives have the following properties for all directions $h_1, h_2 \in L^2(\Omega)$:*

(i) *$S'(u)(h_1) \in H^1_0(\Omega) \cap H^2(\Omega)$ is the solution y' of*

$$Ay' + uy' = -h_1 S(u). \tag{5.1.12}$$

(ii) *$S''(u)(h_1, h_2) \in H^1_0(\Omega) \cap H^2(\Omega)$ is the solution y'' of*

$$Ay'' + uy'' = -h_2 S'(u)(h_1) - h_1 S'(u)(h_2). \tag{5.1.13}$$

(iii) *The following inequalities hold*

$$\|S'(u)(h_1)\| \leq C_2 \|h_1\| \|f\|, \tag{5.1.14}$$

$$\|S''(u)(h_1, h_2)\| \leq C_3 \|h_1\| \|h_2\| \|f\|. \tag{5.1.15}$$

Proof. Part (i) and (ii) can be shown by direct calculation (see [KV09, Lemma 2.9]). So part (iii) is left to be proved. If $y' := S'(u)(h_1) \in H^1_0(\Omega) \cap H^2(\Omega)$ is a solution to

$$Ay' + uy' = -h_1 S(u),$$

for $u \in U_{ad}$ and $f \in L^2(\Omega)$, by using (5.1.10), we obtain

$$\|y'\| \leq C\|y'\|_{L^\infty} \leq \bar{C}\|h_1 S(u)\| \leq \bar{C}\|h_1\| \|S(u)\|_{L^\infty(\Omega)} \leq C_2 \|h_1\| \|f\|, \tag{5.1.16}$$

where the constants depend on the measure of Ω and not on y. Therefore, we obtain (5.1.14) and $y' \in L^\infty(\Omega)$.

Furthermore, let $y'' := S''(u)(h_1, h_2) \in H^1_0(\Omega) \cap H^2(\Omega)$ be a solution to the following problem

$$Ay'' + uy'' = -h_2 S'(u)(h_1) - h_1 S'(u)(h_2),$$

for $u \in U_{ad}$ and $f \in L^2(\Omega)$. With the same argumentation as above and using (5.1.16), we obtain

$$\|y''\| \leq C\|h_2 S'(u)(h_1) + h_1 S'(u)(h_2)\| \leq 2C^3 \|h_1\| \|h_2\| \|f\|.$$

Therefore, we obtain (5.1.15), which completes the proof. □

5.2. Parabolic models

In this section, we discuss parabolic models with linear and bilinear control mechanism.

5.2.1. Linear control mechanism

Consider the following boundary value problem

$$\begin{cases} \partial_t y + Ay + u = f & \text{in } \Omega_T = \Omega \times (0,T) \\ y = y_0 & \text{on } \Omega \times \{t=0\} \\ y = 0 & \text{on } \Sigma = \partial\Omega \times (0,T). \end{cases} \tag{5.2.1}$$

where $\Omega \subset \mathbb{R}^n$ is a bounded domain, $n \leq 3$, $f \in L^2(\Omega_T)$, and $y_0 \in H_0^1(\Omega)$. The operator $\partial_t + A : L^2(0,T;H_0^1(\Omega)) \to L^2(0,T;H^{-1}(\Omega))$ represents a second-order linear parabolic differential operator, where

$$Ay = -\sum_{i,j=1}^{n} \frac{\partial}{\partial x_j}\left(a_{ij}\frac{\partial}{\partial x_i} y\right) + a_0 y,$$

such that $a_{i,j}, a_0 \in L^\infty(\Omega_T)$, and $a_{i,j}$ satisfies the coercivity condition

$$\sum_{i,j=1}^{n} a_{ij}(x,t)\xi_1\xi_2 \geq \theta \sum_{j=1}^{n} \xi_j^2 \quad \text{a.e. in } \Omega_T, \tag{5.2.2}$$

for some $\theta > 0$ and $a_0 \geq 0$.

Here, we define $L^2(0,T;\mathcal{B}) := \{v : (0,T) \to \mathcal{B} \text{ such that } \int_0^T \|v(t)\|_{\mathcal{B}}^2 dt < \infty\}$ for some Banach space $\mathcal{B}$. For the existence and uniqueness of solutions to (5.2.1) see [Eva10, Section 7].

5.2.2. Bilinear control mechanism

Further, we consider the following bilinear parabolic control problem

$$\begin{cases} \partial_t y + Ay + uy = f & \text{in } \Omega_T \\ y = y_0 & \text{on } \Omega \times \{t=0\} \\ y = 0 & \text{on } \Sigma. \end{cases} \tag{5.2.3}$$

In both linear and bilinear control settings, we require $u \in U_{ad}$, with the following set of feasible controls

$$U_{ad} := \{u \in L^2(\Omega_T) : u_a \leq u \leq u_b \text{ a.e. in } \Omega_T\} \subset L^\infty(\Omega_T), \tag{5.2.4}$$

where we choose $u_a \leq 0 \leq u_b$, $u_a < u_b$.

The following theorem from [BA15] ensures existence and uniqueness of (5.2.3).

In the parabolic case, we denote with $\|\cdot\| := \|\cdot\|_{L^2(\Omega_T)}$ for norms in the Hilbert space $L^2(\Omega)$ induced by the inner product $\langle\cdot,\cdot\rangle := \langle\cdot,\cdot\rangle_{L^2(\Omega_T)}$.

Theorem 5.2.1. *[BA15] Suppose that $u \in L^\infty(\Omega_T)$ and $f \in L^2(\Omega_T)$, and the initial condition $y_0 \in H_0^1(\Omega)$. Furthermore, let $\Omega \subset \mathbb{R}^n, n \leq 3$, be a convex and bounded domain with Lipschitz boundary. Then there exists a unique weak solution y to (5.2.3) such that $y \in H^{2,1}(\Omega_T)$, where $H^{2,1}(\Omega_T) = L^2(0,T; H^2(\Omega) \cap H_0^1(\Omega)) \cap H^1(0,T; L^2(\Omega))$ and it fulfills the following inequality*

$$\|y\|_{L^\infty(0,T;L^2(\Omega))} \leq c_1 \left(\|y_0\|_{L^2(\Omega)} + \|f\| \right), \tag{5.2.5}$$

where c_1 depends on u_b. Theorem 5.2.1 ensures the existence of a unique control-to-state operator

$$S : U_{ad} \to H^{2,1}(\Omega_T), \quad u \mapsto S(u). \tag{5.2.6}$$

We note that in the linear case this operator is affine linear, such that its first Fréchet-derivative is independent of u and we have $S'(u)(h) = S(h) + t$ for some constant t independent of u and h. The Fréchet-derivatives of $S(u)$ in the bilinear case have the following properties.

Lemma 5.2.2. *The control-to-state-operator S is at least twice Fréchet-differentiable in U_{ad} with respect to the $L^2(\Omega_T)$-topology and its derivatives have the following properties for all directions $h_1, h_2 \in L^\infty(\Omega_T)$:*

(i) *$S'(u)(h_1) \in H^{2,1}(\Omega_T)$ is the solution y' of*

$$\partial_t y' + Ay' + uy' = -h_1 S(u), \quad y'(\cdot, 0) = 0. \tag{5.2.7}$$

(ii) *$S''(u)(h_1, h_2) \in H^{2,1}(\Omega_T)$ is the solution y'' of*

$$\partial_t y'' + Ay'' + uy'' = -h_2 S'(u)(h_1) - h_1 S'(u)(h_2), \quad y''(\cdot, 0) = 0. \tag{5.2.8}$$

(iii) *The following inequalities hold*

$$\|S'(u)(h_1)\| \leq c_2 \|h_1\| \left(\|f\| + \|y_0\|_{L^2(\Omega)} \right). \tag{5.2.9}$$

$$\|S''(u)(h_1, h_2)\| \leq c_3 \|h_1\| \|h_2\| \left(\|f\| + \|y_0\|_{L^2(\Omega)} \right). \tag{5.2.10}$$

Proof. Part (i) is proven in [BA15], so we sketch the proof of part (ii). From (5.2.5) we have the following estimates for the solution y' of (5.2.7) and y'' of (5.2.8).

$$\|y'\| \leq \|y'\|_{L^2(0,T;H_0^1(\Omega))} \leq C_1 \|h_1\| \leq C_1 \|h_1\|_{L^\infty(\Omega_T)},$$

and therefore

$$\|y''\| \leq C_2 \|h_1\|_{L^\infty(\Omega_T)} \|h_2\|_{L^\infty(\Omega_T)}.$$

It follows that the bilinear mapping $L^\infty(\Omega_T)\times L^\infty(\Omega_T) \to H^{2,1}(\Omega_T) \subset L^2(\Omega_T),\ (h_1,h_2) \mapsto y''$ is continuous with respect to the $L^2(\Omega_T)$-topology. Next, we have that $\tilde{y} = S'(u+h_2)(h_1) - S'(u)(h_1)$ satisfies

$$\partial_t \tilde{y} + A\tilde{y} + u\tilde{y} = -h_1(S(u+h_2) - S(u)) - h_2 S'(u+h_2)(h_1), \quad \tilde{y}(\cdot,0) = 0,$$

such that the following estimate holds

$$\|\tilde{y}\| \le C_3 \|h_1\|_{L^\infty(\Omega_T)} \|h_2\|_{L^\infty(\Omega_T)}.$$

We conclude the proof by an estimate of $w := \tilde{y} - y''$ which satisfies

$$\partial_t w + Aw + uw = -\, h_1(S(u+h_2) - S(u) - S'(u)(h_2)) - h_2\tilde{y}, \quad \tilde{y}(\cdot,0) = 0,$$

such that

$$\|w\| \le C_4 \|h_1\|_{L^\infty(\Omega_T)} \|h_2\|^2_{L^\infty(\Omega_T)}.$$

Thus, we obtain

$$\|S'(u+h_2)(h_1) - S'(u)(h_1) - y''\| \le C \|h_1\|_{L^\infty(\Omega_T)} \|h_2\|^2_{L^\infty(\Omega_T)},$$

which shows that y'' is indeed the Fréchet-derivative of $S'(u)(h_1)$ with respect to the $L^2(\Omega_T)$-topology. Part *(iii)* follows directly from (i), (ii) and (5.2.5). $\square$

6. Optimal control problems with Sparsity Functionals

In this chapter, we discuss optimal control problems governed by the linear- and bilinear-control elliptic and parabolic equations discussed in the previous chapter. We consider the following cost functional

$$J(y,u) := \frac{1}{2}\|y-z\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\|u\|_{L^1}, \tag{6.0.1}$$

where $u \in U_{ad}$, $\alpha, \beta > 0$. Furthermore $z \in L^2(\Omega)$, $y \in H_0^1(\Omega)$ for the elliptic case and $z \in L^2(\Omega_T)$, $y \in H^{2,1}(\Omega_T)$ for the parabolic case.

This functional is made of a Fréchet-differentiable classical tracking type cost with L^2-regularization and a nondifferentiable L^1-control cost. Using the control-to-state operator (5.1.11) in the linear-control and (5.2.6) in the bilinear-control case, we have the following reduced optimal control problem

$$\min_{u \in U_{ad}} \hat{J}(u) := \quad \frac{1}{2}\|S(u)-z\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\|u\|_{L^1}. \tag{6.0.2}$$

6.1. Nonsmooth analysis in function space

For the analysis that follows, we need the definition of derivative for nonconvex, nonsmooth functions that is introduced below.

Definition 6.1.1. *Let X and Y be Banach spaces, $D \subset X$ be open and $\mathcal{F} : D \to Y$ be a nonlinear mapping. We say that $\mathcal{F}$ is generalized differentiable in an open subset $U \subset D$ if there exists a set-valued mapping $\partial^*\mathcal{F} : D \rightrightarrows \mathcal{L}(X,Y)$ with $\partial^*\mathcal{F}(x) \neq \emptyset$ for all $x \in D$ such that*

$$\lim_{h\to 0} \frac{1}{\|h\|_X}\|\mathcal{F}(x+h) - \mathcal{F}(x) - \mathcal{G}(x+h)h\|_Y = 0, \tag{6.1.1}$$

for every $\mathcal{G} \in \partial^*\mathcal{F}$ and for every $x \in U$. We call $\partial^*\mathcal{F}$ the generalized differential and every $\mathcal{G} \in \partial^*\mathcal{F}$ a generalized derivative.

This definition is similar to the semismoothness stated in [Ulb11] and also known under the name 'slant differentiability'; see, e.g., [CNQ00].

For convex functionals on Hilbert spaces, the generalized differential is equivalent to the following subdifferential.

Lemma 6.1.1. *Let H be a Hilbert space, $D \subset H$ be open and $F : D \to \mathbb{R}$ be a convex functional. The mapping $\partial\mathcal{F} : H \rightrightarrows H^*$ given by*

$$\partial\mathcal{F}(x) := \{\gamma \in H^* : \langle\gamma, y - x\rangle_{H^*,H} \leq \mathcal{F}(y) - \mathcal{F}(x) \;\; \textit{for all } y \in H\}$$

is called the subdifferential of $\mathcal{F}$ in u and it holds $\partial^*\mathcal{F} = \partial\mathcal{F}$.

Proof.

'$\subset$': Let $\gamma \in \partial^*\mathcal{F}$. We first show that the mapping $g : (0, 1] \to \mathbb{R}$ defined by

$$g(t) := \frac{\mathcal{F}(x + t(y - x)) - \mathcal{F}(x)}{t}.$$

is monotonically increasing. Therefore, consider t_1, t_2 with $0 < t_1 < t_2 < 1$ and define $t' := \frac{t_1}{t_2} \in (0, 1)$ and $z = x + t_2(y - x)$. Due to the convexity of $\mathcal{F}$, it holds that

$$\mathcal{F}(x + t'(z - x)) \leq t'\mathcal{F}(z) + (1 - t')\mathcal{F}(x).$$

Inserting $t' := \frac{t_1}{t_2} \in (0, 1)$ and $z = x + t_2(y - x)$ yields

$$\frac{\mathcal{F}(x + t_1(y - x)) - \mathcal{F}(x)}{t_1} \leq \frac{\mathcal{F}(x + t_2(y - x)) - \mathcal{F}(x)}{t_2},$$

and, hence, that g is monotonically increasing. Furthermore, by replacing $h = t(y - x)$, one has that

$$\begin{aligned} 0 &= \lim_{h\to 0} \frac{1}{\|h\|_H} |\mathcal{F}(x + h) - \mathcal{F}(x) - \langle\gamma(x + h), h\rangle| \\ &= \lim_{t\to 0} \frac{|\mathcal{F}(x + t(y - x)) - \mathcal{F}(x) - t\,\langle\gamma(x + t(y - x)), y - x\rangle|}{t\|y - x\|} \\ &= \lim_{t\to 0} \frac{|g(t) - \langle\gamma(x + t(y - x)), y - x\rangle|}{\|y - x\|}. \end{aligned}$$

Hence,

$$\langle\gamma(x), y - x\rangle = \lim_{t\to 0} \langle\gamma(x + t(y - x)), y - x\rangle = \lim_{t\to 0} g(t) \leq g(1) = \mathcal{F}(y) - \mathcal{F}(x).$$

'$\supset$': Let $\gamma \in \partial \mathcal{F}$. Then, $\langle \gamma(x), y-x \rangle \leq \mathcal{F}(y) - \mathcal{F}(x)$, and we have

$$\begin{aligned}
&\lim_{h \to 0} \frac{1}{\|h\|_H} |\mathcal{F}(x+h) - \mathcal{F}(x) - \langle \gamma(x+h), h \rangle | \\
=& \lim_{y \to x} \frac{1}{\|y-x\|_H} |\mathcal{F}(y) - \mathcal{F}(x) - \langle \gamma(y), y-x \rangle | \\
=& \lim_{y \to x} \frac{1}{\|y-x\|_H} \Big(\langle \gamma(y), y-x \rangle + \mathcal{F}(x) - \mathcal{F}(y) \Big) \\
=& \lim_{t \to 0} \frac{\Big(\langle \gamma(x+t(y-x)), t(y-x) \rangle + \mathcal{F}(x) - \mathcal{F}(x+t(y-x)) \Big)}{t\|y-x\|} \\
=& \frac{1}{\|y-x\|_H} \left(\langle \gamma(x), y-x \rangle - \lim_{t \to 0} \frac{\mathcal{F}(x+t(y-x)) - \mathcal{F}(x)}{t} \right) \\
\leq& \frac{1}{\|y-x\|_H} \left(\langle \gamma(x), y-x \rangle - \lim_{t \to 0} \frac{t \, \langle \gamma(x), y-x \rangle}{t} \right) = 0,
\end{aligned}$$

where the third equality is due to the fact, that $\mathcal{F}(y) - \mathcal{F}(x) - \langle \gamma(y), y-x \rangle = \langle \gamma(y), x-y \rangle - (\mathcal{F}(x) - \mathcal{F}(y)) \leq 0$. Hence, $\gamma \in \partial^* \mathcal{F}$. □

6.2. Convexity of the cost functional

The nondifferentiable part $\hat{J}_1(u) := \beta \|u\|_{L^1}$ is convex. Therefore, in order to discuss local convexity of the reduced functional $\hat{J}(u)$, we investigate the second derivative of the differentiable part $\hat{J}_2(u) := \frac{1}{2}\|S(u) - z\|^2 + \frac{\alpha}{2}\|u\|^2$. We have

$$\hat{J}_2''(\bar{u})(v,w) = \langle S'(\bar{u})(v), S'(\bar{u})(w) \rangle + \langle S(\bar{u}) - z, S''(\bar{u})(v,w) \rangle + \alpha \langle v, w \rangle .$$

In particular, in the linear case, we have

$$\hat{J}_2''(\bar{u})(v,v) = \|S'(u)(v)\|^2 + \alpha \|v\|^2 > 0, \text{ for all } v \in L^2(\Omega), \; \|v\| \neq 0. \tag{6.2.1}$$

We conclude that the reduced functional is strictly convex in the linear case.

In the bilinear case, we have a non-convex optimization problem. However, local convexity can be guaranteed under some conditions. To be specific, we chose the sufficient condition stated in the following theorem.

Lemma 6.2.1. *Let $C''(u) := \sup_{\|v\| \leq 1} \|S''(u)(v,v)\|$, if the following inequality holds*

$$C''(u)\|S(u) - z\| < \alpha, \tag{6.2.2}$$

then the reduced functional $\hat{J}(u)$ is strictly convex in a neighborhood of $u \in U_{ad}^{bil}$.

Proof. Since $\hat{J}_1(u) := \beta \|u\|_{L^1}$ is convex, we have to prove that $\hat{J}_2(u) := \frac{1}{2}\|S(u) - z\|^2 + \frac{\alpha}{2}\|u\|^2$ is strictly convex in u. Therefore we show that the reduced Hessian is positive definite in U_{ad} as follows

$$\begin{aligned}\hat{J}_2''(u)(v,v) &= \langle S'(u)(v), S'(u)(v)\rangle + \langle S(u)-z, S''(u)(v,v)\rangle + \alpha\,\langle v,v\rangle \\ &\geq (\alpha - C''(u)\|S(u)-z\|)\|v\|^2,\end{aligned}$$

and thus $\hat{J}(u)$ is strictly convex in u. □

We remark that the result of Lemma 6.2.1 is well known. It expresses local convexity of the reduced objective when the state function is sufficiently close to the target and the weight of the quadratic L^2 cost of the control is sufficiently large. Indeed, local convexity may result with much weaker assumptions. However, for the investigation of the fast proximal schemes (FTIP), see Chapter 7, we need strict convexity of the cost functional. Therefore, we make the following strong assumption that is required in the formulation of the FTIP method.

Assumption 1. *We assume that (6.2.2) holds for all $u \in U_{ad}$.*

Remark 6.2.1. *Because of Lemma 5.1.4, this assumption holds if the regularization parameter $\alpha > C_3\|f\|(C\|f\| + \|z\|)$ for the elliptic case and because of Lemma 5.2.2 it holds for $\alpha > c_3\left(\|f\| + \|y_0\|_{L^2(\Omega)}\right)\left[c_1\left(\|f\| + \|y_0\|_{L^2(\Omega)}\right) + \|z\|\right]$ for the parabolic case.*

6.3. Optimality conditions

To characterize the optimal control sought, we discuss in the following the first-order optimality conditions.

6.3.1. Elliptic models

In this section, we investigate the convexity conditions and optimality conditions for (6.0.2), where S(u) is the control-to-state operator of the elliptic model (5.1.11).

In the next step, the optimality conditions of (6.0.2) are derived. From [ET99, Remark 3.2], we obtain that $\bar{u}$ is a solution of (6.0.2) if and only if there exists a $\bar{\lambda} \in \partial\hat{J}_1(\bar{u})$ such that

$$\left\langle S'(\bar{u})^*(S(\bar{u})-z) + \alpha\bar{u} + \bar{\lambda}, u - \bar{u}\right\rangle \geq 0, \quad \text{for all } u \in U_{ad}, \tag{6.3.1}$$

where * denotes the adjoint operator. From (6.3.1) one can derive the optimality system by using the Lagrange multipliers $\bar{\lambda}_a$, $\bar{\lambda}_b \in L^2(\Omega)$ (see [Sta09, Theorem 2.1]):

Theorem 6.3.1. *The optimal solution $\bar{u}$ of (6.0.2) is characterized by the existence of*

$(\bar{\lambda}, \bar{\lambda}_a, \bar{\lambda}_b) \in L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega)$ *such that*

$$S'(\bar{u})^*(S(\bar{u}) - z) + \alpha\bar{u} + \bar{\lambda} + \bar{\lambda}_b - \bar{\lambda}_a = 0, \tag{6.3.2}$$
$$\bar{\lambda}_b \geq 0, \quad u_b - \bar{u} \geq 0, \quad \bar{\lambda}_b(u_b - \bar{u}) = 0, \tag{6.3.3}$$
$$\bar{\lambda}_a \geq 0, \quad \bar{u} - u_a \geq 0, \quad \bar{\lambda}_a(\bar{u} - u_a) = 0, \tag{6.3.4}$$
$$\bar{\lambda} = \beta \quad a.e. \ on \ \{x \in \Omega : \bar{u} > 0\}, \tag{6.3.5}$$
$$|\bar{\lambda}| \leq \beta \quad a.e. \ on \ \{x \in \Omega : \bar{u} = 0\}, \tag{6.3.6}$$
$$\bar{\lambda} = -\beta \quad a.e. \ on \ \{x \in \Omega : \bar{u} < 0\}. \tag{6.3.7}$$

If one introduces the parameter $\bar{\mu} := \bar{\lambda} + \bar{\lambda}_b - \bar{\lambda}_a$, it is shown in [Sta09] that the conditions (6.3.3)-(6.3.7) are equivalent to

$$B(\bar{u}, \bar{\mu}) = 0, \tag{6.3.8}$$

where

$$\begin{aligned} B(\bar{u}, \bar{\mu}) :=& \bar{u} - \max\{0, \bar{u} + c(\bar{\mu} - \beta)\} - \min\{0, \bar{u} + c(\bar{\mu} + \beta)\} \\ &+ \max\{0, \bar{u} - u_b + c(\bar{\mu} - \beta)\} + \min\{0, \bar{u} - u_a + c(\bar{\mu} + \beta)\}, \end{aligned}$$

where $c > 0$ is arbitrary. With this setting (6.3.2)-(6.3.7) reduces to

$$S'(\bar{u})^*(S(\bar{u}) - z) + \alpha\bar{u} + \bar{\mu} = 0, \tag{6.3.9}$$
$$B(\bar{u}, \bar{\mu}) = 0. \tag{6.3.10}$$

Next we discuss the linear control mechanism (5.1.1). In the linear control case, the equation (6.3.9) becomes the following

$$-A^{-*}(A^{-1}(f - \bar{u}) - z) + \alpha\bar{u} + \bar{\mu} = 0, \tag{6.3.11}$$

where $A^{-*} = (A^*)^{-1}$. By setting $\bar{y} = A^{-1}(f - \bar{u})$ and $\bar{p} := -A^{-*}(\bar{y} - z)$ equation (6.3.11) can be written as follows

$$\bar{p} + \alpha\bar{u} + \bar{\mu} = 0.$$

We summarize the previous considerations into the following theorem.

Theorem 6.3.2. *(Linear optimality conditions) The optimal solution $(\bar{y}, \bar{u}) \in H_0^1(\Omega) \times L^2(\Omega)$ to (6.0.2) in the linear control case is characterized by the existence of the dual pair $(\bar{p}, \bar{\mu}) \in H_0^1(\Omega) \times L^2(\Omega)$ such that*

$$A\bar{y} + \bar{u} - f = 0 \tag{6.3.12}$$
$$A^*\bar{p} + \bar{y} - z = 0 \tag{6.3.13}$$
$$\bar{p} + \alpha\bar{u} + \bar{\mu} = 0 \tag{6.3.14}$$
$$B(\bar{u}, \bar{\mu}) = 0. \tag{6.3.15}$$

Furthermore, the explicit gradient and the Hessian of $\hat{J}_2(u)$ are given by

$$\nabla \hat{J}_2(u) = \alpha u + p \tag{6.3.16}$$

and

$$\nabla^2 \hat{J}_2(u) = \alpha I + A^{-*}A^{-1} \tag{6.3.17}$$

Next, we discuss the bilinear elliptic control mechanism (5.1.3). For the bilinear system, we have $S'(u)(h_1) = -(A+u)^{-1}[h_1(A+u)^{-1}f]$ and therefore $S'(u)^*(h_1) = -(A+u)^{-1}f(A+u)^{-*}h_1$ such that equation (6.3.9) becomes the following

$$-(A+\bar{u})^{-1}f(A+\bar{u})^{-*}((A+\bar{u})^{-1}f - z) + \alpha\bar{u} + \bar{\mu} = 0. \tag{6.3.18}$$

By setting $\bar{y} = (A+\bar{u})^{-1}f$ and $\bar{p} := -(A+\bar{u})^{-*}(\bar{y}-z)$ this can be written as follows

$$\bar{y}\bar{p} + \alpha\bar{u} + \bar{\mu} = 0.$$

We summarize the previous considerations into the following theorem.

Theorem 6.3.3. *(Bilinear optimality system) The optimal solution $(\bar{y},\bar{u}) \in H_0^1(\Omega) \times L^2(\Omega)$ to (6.0.2) in the bilinear control case is characterized by the existence of the dual pair $(\bar{p},\bar{\mu}) \in H_0^1(\Omega) \times L^2(\Omega)$ such that*

$$\begin{aligned} &A\bar{y} + \bar{u}\bar{y} - f = 0 \\ &A^*\bar{p} + \bar{y} + \bar{u}\bar{p} - z = 0 \\ &\bar{y}\bar{p} + \alpha\bar{u} + \bar{\mu} = 0 \\ &B(\bar{u},\bar{\mu}) = 0. \end{aligned} \tag{6.3.19}$$

Furthermore, the explicit gradient and the Hessian of $\hat{J}_2(u)$ are given by

$$\nabla \hat{J}_2(u) = \alpha u + py \tag{6.3.20}$$

and

$$\hat{J}_2''(u)(v_1,v_2) = \left\langle v_1, \nabla^2 \hat{J}_2(u) v_2 \right\rangle, \tag{6.3.21}$$

where

$$\nabla^2 \hat{J}_2(u)(\cdot) = y(A+u)^{-*}(A+u)^{-1}(y(\cdot)) - y(A+u)^{-*}(p(\cdot)) - p(A+u)^{-1}(y(\cdot)) + \alpha(\cdot).$$

6.3.2. Parabolic models

In this section, we investigate the convexity conditions and optimality conditions for (6.0.2), where S(u) is the control-to-state operator of the parabolic model (5.2.6).

In the next step the optimality conditions of (6.0.2) are derived. From [ET99, Remark 3.2] we obtain that $\bar{u}$ is a solution of (6.0.2) if and only if there exists a $\bar{\lambda} \in \partial\hat{J}_1(\bar{u})$ such that

$$\left\langle S'(\bar{u})^*(S(\bar{u}) - z) + \alpha\bar{u} + \bar{\lambda}, u - \bar{u} \right\rangle \geq 0, \quad \text{for all } u \in U_{ad}, \tag{6.3.22}$$

where * denotes the adjoint operator. From (6.3.22) one can derive the optimality system by using the Lagrange multipliers $\bar{\lambda}_a$, $\bar{\lambda}_b \in L^\infty(\Omega_T)$ (see [Sta09, Theorem 2.1]):

Theorem 6.3.4. *The optimal solution $\bar{u}$ of (6.0.2) is characterized by the existence of $(\bar{\lambda}, \bar{\lambda}_a, \bar{\lambda}_b) \in L^2(\Omega_T) \times L^\infty(\Omega_T) \times L^\infty(\Omega_T)$ such that*

$$S'(\bar{u})^*(S(\bar{u}) - z) + \alpha\bar{u} + \bar{\lambda} + \bar{\lambda}_b - \bar{\lambda}_a = 0, \tag{6.3.23}$$
$$\bar{\lambda}_b \geq 0, \quad u_b - \bar{u} \geq 0, \quad \bar{\lambda}_b(u_b - \bar{u}) = 0, \tag{6.3.24}$$
$$\bar{\lambda}_a \geq 0, \quad \bar{u} - u_a \geq 0, \quad \bar{\lambda}_a(\bar{u} - u_a) = 0, \tag{6.3.25}$$
$$\bar{\lambda} = \beta \quad a.e.\ on\ \{x \in \Omega : \bar{u} > 0\}, \tag{6.3.26}$$
$$\bar{\lambda} \leq \beta \quad a.e.\ on\ \{x \in \Omega : \bar{u} = 0\}, \tag{6.3.27}$$
$$\bar{\lambda} = -\beta \quad a.e.\ on\ \{x \in \Omega : \bar{u} < 0\}. \tag{6.3.28}$$

If one introduces the parameter $\bar{\mu} := \bar{\lambda} + \bar{\lambda}_b - \bar{\lambda}_a$, it is shown in [Sta09] that conditions (6.3.24)-(6.3.28) are equivalent to

$$B(\bar{u}, \bar{\mu}) = 0, \tag{6.3.29}$$

where we define

$$\begin{aligned} B(\bar{u}, \bar{\mu}) :=& \bar{u} - \max\{0, \bar{u} + c(\bar{\mu} - \beta)\} - \min\{0, \bar{u} + c(\bar{\mu} + \beta)\} \\ &+ \max\{0, \bar{u} - u_b + c(\bar{\mu} - \beta)\} + \min\{0, \bar{u} - u_a + c(\bar{\mu} + \beta)\}. \end{aligned}$$

With this setting (6.3.23)-(6.3.28) reduces to

$$S'(\bar{u})^*(S(\bar{u}) - z) + \alpha\bar{u} + \bar{\mu} = 0, \tag{6.3.30}$$
$$B(\bar{u}, \bar{\mu}) = 0. \tag{6.3.31}$$

For the linear parabolic problem (5.2.1), we define $\bar{y} := S(\bar{u})$ and introduce the adjoint operator $\bar{p}$ as a solution to

$$-\partial_t\bar{p} + A^*\bar{p} + \bar{y} - z = 0, \quad \bar{p}(\cdot, T) = 0$$

From standard arguments, e.g., [Trö09], we see that equation (6.3.30) can be written as follows

$$\bar{p} + \alpha\bar{u} + \bar{\mu} = 0$$

We summarize the previous considerations into the following theorem.

Theorem 6.3.5. *(Optimality conditions for the parabolic linear control problem) The optimal solution $(\bar{y}, \bar{u}) \in H^{2,1}(\Omega_T) \times L^\infty(\Omega_T)$ to (6.0.2), in the linear control case, is characterized by the existence of the dual pair $(\bar{p}, \bar{\mu}) \in H^{2,1}(\Omega_T) \times L^\infty(\Omega_T)$ such that*

$$\partial_t\bar{y} + A\bar{y} + \bar{u} - f = 0 \qquad in\ \Omega_T \tag{6.3.32}$$
$$-\partial_t\bar{p} + A^*\bar{p} + \bar{y} - z = 0 \qquad in\ \Omega_T \tag{6.3.33}$$
$$\bar{p} + \alpha\bar{u} + \bar{\mu} = 0 \qquad in\ \Omega_T \tag{6.3.34}$$
$$B(\bar{u}, \bar{\mu}) = 0 \qquad in\ \Omega_T \tag{6.3.35}$$
$$\bar{y} = y_0 \qquad on\ \Omega \times \{t = 0\} \tag{6.3.36}$$
$$\bar{p} = 0 \qquad on\ \Omega \times \{t = T\} \tag{6.3.37}$$

Furthermore, in the linear control case, the explicit gradient of $\hat{J}_2(u)$ is given by

$$\nabla \hat{J}_2(u) = \alpha u + p \tag{6.3.38}$$

For the bilinear parabolic problem (5.2.3), we define $\bar{y} = S(\bar{u})$ and introduce the adjoint operator $\bar{p}$ as a solution of

$$-\partial_t \bar{p} + A^* \bar{p} + \bar{u}\bar{p} + \bar{y} - z = 0, \quad \bar{p}(\cdot, T) = 0.$$

From standard arguments, e.g., [Trö09] we see that equation (6.3.30) can be written as

$$\bar{y}\bar{p} + \alpha \bar{u} + \bar{\mu} = 0.$$

We summarize the previous considerations into the following theorem.

Theorem 6.3.6. *(Optimality conditions for the parabolic bilinear control problem) The optimal solution $(\bar{y}, \bar{u}) \in H^{2,1}(\Omega_T) \times L^\infty(\Omega_T)$ to (6.0.2), in the bilinear control case, is characterized by the existence of the dual pair $(\bar{p}, \bar{\mu}) \in H^{2,1}(\Omega_T) \times L^\infty(\Omega_T)$ such that*

$$\partial_t \bar{y} + A\bar{y} + \bar{u}\bar{y} - f = 0 \qquad \text{in } \Omega_T \tag{6.3.39}$$

$$-\partial_t \hat{p} + A^* \bar{p} + \bar{y} + \bar{u}\bar{p} - z = 0 \qquad \text{in } \Omega_T \tag{6.3.40}$$

$$\bar{y}\bar{p} + \alpha \bar{u} + \bar{\mu} = 0 \qquad \text{in } \Omega_T \tag{6.3.41}$$

$$B(\bar{u}, \bar{\mu}) = 0 \qquad \text{in } \Omega_T \tag{6.3.42}$$

$$\bar{y} = y_0 \qquad \text{on } \Omega \times \{t = 0\} \tag{6.3.43}$$

$$\bar{p} = 0 \qquad \text{on } \Omega \times \{t = T\} \tag{6.3.44}$$

Furthermore, in the bilinear control case, the explicit gradient of $\hat{J}_2(u)$ is given by

$$\nabla \hat{J}_2(u) = \alpha u + py \tag{6.3.45}$$

7. Proximal methods in function spaces

In this section, we discuss first-order inertial proximal methods to solve our linear and bilinear optimal control problems. The starting point to discuss proximal methods consists of identifying a smooth and a nonsmooth part in the reduced objective $\hat{J}(u)$. That is, we consider the following optimization problem

$$\min_{u \in U_{ad}} \hat{J}(u) := \hat{J}_1(u) + \hat{J}_2(u), \tag{7.0.1}$$

where we assume

$$\hat{J}_1(u) \text{ is continuous, convex and nondifferentiable} \tag{7.0.2}$$

$$\begin{aligned} &\hat{J}_2(u) \text{ is Q-differentiable with respect to } U_{ad}, \\ &\qquad \text{and has Lipschitz-continuous gradient:} \\ &\qquad \|\nabla \hat{J}_2(u) - \nabla \hat{J}_2(v)\| \leq L(\hat{J}_2)\|u - v\|, \quad \forall u, v \in U_{ad}, \end{aligned} \tag{7.0.3}$$

where $L(\hat{J}_2) > 0$. The following lemma is essential in the formulation of proximal methods.

Lemma 7.0.1. *Let J_2 be Q-differentiable with respect to U_{ad} and it has Lipschitz continuous gradient with Lipschitz constant $L(J_2)$ (7.0.3). Then for all $L \geq L(J_2)$, the following holds.*

$$\hat{J}_2(u) \leq \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \frac{L}{2}\|u - v\|^2, \quad \forall u, v \in U_{ad}. \tag{7.0.4}$$

Proof.

$$\begin{aligned} \hat{J}_2(u) &= \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \int_0^1 \left\langle \nabla \hat{J}_2(v + t(u - v)) - \nabla \hat{J}_2(v), u - v \right\rangle \mathrm{dt} \\ &\leq \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \int_0^1 \left\| \nabla \hat{J}_2(v + t(u - v)) - \nabla \hat{J}_2(v) \right\| \|u - v\| \mathrm{dt} \\ &\leq \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \int_0^1 Lt\|u - v\|^2 \mathrm{dt} \\ &\leq \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \frac{L}{2}\|u - v\|^2. \end{aligned}$$

$\square$

Notice that $L := L(\hat{J}_2)$ represents the smallest value of L such that (7.0.4) is satisfied. We remark that the discussion that follows is valid for $L \geq L(\hat{J}_2)$ as in (7.0.4). However, as we discuss below, the efficiency of our proximal schemes depends on how close is the chosen L to the minimal and optimal value $L(\hat{J}_2)$. Now, since this value is usually not available analytically, we discuss and implement below some numerical strategies for determining a sufficiently accurate approximation of $L(\hat{J}_2)$. In particular, we consider a power iteration [Wil88], and the backtracking approach discussed in Algorithm 8. Further, notice that also in the case of choosing $L >> L(\hat{J}_2)$, our proximal scheme still converges with the same convergence rate as shown in Section 7.3. However, the convergence constant grows considerably as L becomes larger and therefore the convergence of the proximal method appears recognizably slower. On the other hand, if L is chosen smaller than the Lipschitz constant, then convergence cannot be guaranteed.

The strategy of the proximal scheme is to minimize an upper bound of the objective functional at each iteration, instead of minimizing the functional directly. Lemma 7.0.1 gives us the following upper bound for all $v \in U_{ad}$. We have

$$\min_{u\in U_{ad}} \left\{\hat{J}_1(u) + \hat{J}_2(u)\right\} \leq \min_{u\in U_{ad}} \left\{\hat{J}_1(u) + \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(y), u - v\right\rangle + \frac{L}{2}\|u-v\|^2\right\},$$

where we have equality if $u = v$. Furthermore, we have the following equation

$$\begin{aligned}&\arg\min_{u\in U_{ad}} \left\{\hat{J}_1(u) + \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v\right\rangle + \frac{L}{2}\|u-v\|^2\right\}\\ =&\arg\min_{u\in U_{ad}} \left\{\hat{J}_1(u) + \frac{L}{2}\left\|u - \left(v - \frac{1}{L}\nabla \hat{J}_2(v)\right)\right\|^2\right\}.\end{aligned} \tag{7.0.5}$$

In the optimal control problems stated in Chapter 6, we have $\hat{J}_1(u) = \beta\|u\|_{L^1}$ and (7.0.5) has an explicit solution, that we discuss in the following lemma.

Lemma 7.0.2. *The following equation holds*

$$\arg\min_{u\in U_{ad}} \left\{\tau\|u\|_{L^1} + \frac{1}{2}\|u-v\|^2\right\} = \mathbb{S}_\tau^{U_{ad}}(v) \quad \text{for any } v \in L^2(\Omega),$$

where the projected soft thresholding function is defined as follows

$$\mathbb{S}_\tau^{U_{ad}}(v) := \begin{cases} \min\{v-\tau, u_b\} & \text{on } \{x\in\Omega : v(x) > \tau\} \\ 0 & \text{on } \{x\in\Omega : |v(x)| \leq \tau\} \\ \max\{v+\tau, u_a\} & \text{on } \{x\in\Omega : v(x) < -\tau\} \end{cases}.$$

Proof. There exists a $\gamma(\bar{u}) \in \partial\|\bar{u}\|_{L^1}$, the subdifferential of $\|\cdot\|_{L_1}$ such that the solution $\bar{u} := \arg\min_{u\in U_{ad}} \left\{\tau\|u\|_{L^1} + \frac{1}{2}\|u-v\|^2\right\}$ fulfills the following variational inequality; see, e.g., [ET99];

$$\langle \bar{u} - v + \tau\gamma(\bar{u}), u - \bar{u}\rangle \geq 0, \quad \forall u \in U_{ad}. \tag{7.0.6}$$

Now, we show that $\hat{u} := \mathbb{S}_\tau^{U_{ad}}(v)$ fulfills (7.0.6). The following investigation of the different cases is meant to be pointwise. We have

- $v - \tau > u_b \geq 0$:
  It follows that $\hat{u} = u_b$ and therefore $\gamma(\hat{u}) = 1$ such that
  $(u_b - v + \tau)(u - u_b) \geq 0, \ \forall u \in U_{ad}$.

- $0 < v - \tau < u_b$:
  It follows that $\hat{u} = v - \tau > 0$ and $\gamma(\hat{u}) = 1$ such that
  $(\hat{u} - v + \tau)(u - u_b) = 0, \ \forall u \in U_{ad}$.

- $|v| \leq \tau$:
  It follows that $\hat{u} = 0$ and $\gamma(\hat{u}) = \frac{v}{\tau} \in B_1(0)$ such that
  $\left(\hat{u} - v + \tau\left(\frac{v}{\tau}\right)\right)(u - \hat{u}) = 0, \ \forall u \in U_{ad}$.

- $u_a < v + \tau < 0$:
  It follows that $\hat{u} = v + \tau < 0$ and therefore $\gamma(\hat{u}) = -1$ such that
  $\langle \hat{u} - v - \tau, u - \hat{u} \rangle = 0, \ \forall u \in U_{ad}$

- $v + \tau < u_a \leq 0$:
  It follows that $\hat{u} = u_a$ and therefore $\gamma(\hat{u}) = -1$ such that
  $(u_a - v - \tau)(u - u_a) \geq 0, \ \forall u \in U_{ad}$. □

Based on this lemma, we conclude that the solution to (7.0.5) is given by

$$\underset{u \in U_{ad}}{\arg\min} \left\{ \hat{J}_1(u) + \frac{L}{2} \left\| u - \left(v - \frac{1}{L} \nabla \hat{J}_2(v) \right) \right\|^2 \right\} = \mathbb{S}^{U_{ad}}_{\frac{\beta}{L}} \left(v - \frac{1}{L} \nabla \hat{J}_2(v) \right),$$

thus obtaining an approximation to the optimal u sought. Therefore we can use this result to define a general iterative scheme as follows

$$u_{k+1} \leftarrow \mathbb{S}^{U_{ad}}_{\beta \cdot s_k} \left(u_k - s_k \nabla \hat{J}_2(u_k) + \theta_k \|u_k - u_{k-1}\| \right), \tag{7.0.7}$$

starting from given $u_0 = u_{-1}$. For $s_k := \frac{1}{L}$ and $\theta_k = 0$ we have the iterative scheme discussed above that. We investigate requirements on the steplength s_k and the inertial parameter θ_k such that the general method provides convergence towards a solution of our optimal control problem.

The update step (7.0.7) requires the solution of (6.3.12) and (6.3.13), resp. (6.3.39) and (6.3.40), to get y and p for the calculation of

$$\begin{aligned} \nabla \hat{J}_2(u) &= p + \alpha u \ \ \text{(linear)} \\ \text{resp.} \quad \nabla \hat{J}_2(u) &= py + \alpha u \ \ \text{(bilinear)}. \end{aligned}$$

However, the exact inversion of a discretized differential operator $\mathcal{A} := A$ in the elliptic case resp. $\mathcal{A} := \partial_t + A$ in the parabolic case, may become too expensive. Therefore one has to estimate an approximate solution; e.g., the conjugate gradient method [HS52]. For this reason, we discuss a truncated version of the proximal scheme where the equality constraints and the corresponding adjoint equations are solved up to a given tolerance quantified by $\varepsilon > 0$. In the following, we denote by $\nabla_\varepsilon \hat{J}_2(u)$ the truncated gradient that corresponds to a truncated integration of the equation $\mathcal{A}y = f - u$, resp. $(\mathcal{A} + u)y = f$, that results in an approximated state variable y^ε, resp. p^ε, in the following sense

$$\|\mathcal{A}y^\varepsilon - f + u\| \leq \varepsilon, \qquad \text{resp.} \qquad \|\mathcal{A}y^\varepsilon + uy^\varepsilon - f\| \leq \varepsilon.$$

Hence, there exists an $\tilde{e} \in L^2(\Omega)$ with $\|\tilde{e}\| < \varepsilon$ such that

$$\mathcal{A}y^\varepsilon = f - u + \tilde{e}, \qquad \text{resp.} \qquad \mathcal{A}y^\varepsilon + uy^\varepsilon = f + \tilde{e}. \tag{7.0.8}$$

We denote the truncated inversion method for the problem $By = g$, with an error $\|By^\varepsilon - g\| \leq \varepsilon$, with $inv(B, g, \varepsilon)$. With this notation, the truncated gradient computation is illustrated in Algorithm 5 and 6.

Algorithm 5 (Calculation of the truncated gradient $\nabla_\varepsilon \hat{J}_2(u)$) – elliptic case

Require: A, f, z, ε, u

1. $y^\varepsilon = inv(A, f - u, \varepsilon)$, resp. $y^\varepsilon = inv(A + u, f, \varepsilon)$
2. $p^\varepsilon = inv(A^*, z - y^\varepsilon, \varepsilon)$, resp. $p^\varepsilon = inv(A^* + u, z - y^\varepsilon, \varepsilon)$
3. $\nabla_\varepsilon \hat{J}_2(u) = p^\varepsilon + \alpha u$, resp. $\nabla_\varepsilon \hat{J}_2(u) = p^\varepsilon y^\varepsilon + \alpha u$

Algorithm 6 (Calculation of the truncated gradient $\nabla_\varepsilon \hat{J}_2(u)$) – parabolic case

Require: A, f, z, ε, u

1. $y^\varepsilon = inv(\partial_t + A, f - u, \varepsilon)$, resp. $y^\varepsilon = inv(\partial_t + A + u, f, \varepsilon)$
2. $p^\varepsilon = inv(-\partial_t + A^*, z - y^\varepsilon, \varepsilon)$, resp. $p^\varepsilon = inv(-\partial_t + A^* + u, z - y^\varepsilon, \varepsilon)$
3. $\nabla_\varepsilon \hat{J}_2(u) = p^\varepsilon + \alpha u$, resp. $\nabla_\varepsilon \hat{J}_2(u) = p^\varepsilon y^\varepsilon + \alpha u$

7.1. Inertial proximal algorithms

With this preparation, we formulate our general truncated inertial proximal schemes given by Algorithms 7 8 & 9.

Algorithm 7 (General truncated inertial proximal (GTIP) method)

Require: β, $\hat{J}_2$, $u_0 = u_{-1}$, U_{ad}, TOL, $c_1, c_2, c_3 > 0$ close to 0;
Initialize: $B_0 = 1$, $k = 0$;
while $\|B_{k-1}\| > TOL$ **do**
1. $u_{k+1} \leftarrow \mathbb{S}^{U_{ad}}_{\beta \cdot s_k}\left(u_k - s_k \nabla \hat{J}_2^{\varepsilon_k}(u_k) + \theta_k(u_k - u_{k-1})\right)$
where $s_k \geq c_1$, $\theta_k \geq 0$ are chosen such that $\delta_k \geq \gamma_k \geq c_2 + c_3$, defined by

$$\delta_k := \frac{1}{s_k} - \frac{L_n}{2} - \frac{\theta_k}{2s_k} \quad \text{and} \quad \gamma_k := \frac{1}{s_k} - \frac{L_n}{2} - \frac{\theta_k}{s_k}$$

with L_k satisfying

$$\hat{J}_2(u_{k+1}) \leq \hat{J}_2(u_k) + \left\langle \nabla \hat{J}_2(u_k), u_{k+1} - u_k \right\rangle + \frac{L_k}{2}\|u_{k+1} - u_k\|^2,$$

$\varepsilon_k \leq (\gamma_k - c_2)\frac{\|u_k - u_{k-1}\|^2}{c(u_b - u_a)}$, $c > 0$ is the constant defined in Lemma 7.4.4, and $(\delta_k)_k$ is monotonically decreasing.
2. $\mu_k = -\alpha u_k - S'(u_k)^*(S(u_k) - z)$ (6.3.9)
3. $B_k = B(u_k, \mu_k)$
4. $k = k + 1$

end while

This scheme is discussed in [OCBP14] for the case of finite-dimensional optimization problems. The convergence results for Algorithm 7 presented in [OCBP14] can be extended to our linear and bilinear parabolic control problems.

The following algorithm is a special case of Algorithm 7 in the case that it is possible to calculate the a priori Lipschitz constant of the gradient directly.

Algorithm 8 (Constant truncated inertial proximal (CTIP) method)

Require: β, $\hat{J}_2$, $u_0 = u_{-1}$, U_{ad}, TOL, ε_0;
Initialize: Set $B_0 = 1$, $k = 0$, choose $\theta \in [0, 1)$ and some small $c_2 > 0$;
Calculate the Lipschitz constant $L(\hat{J}_2) = \lambda_{max}(\nabla^2 \hat{J}_2)$ and set $s < 2(1-\theta)/(L + 2c_2)$.
while $\|B_k\| > TOL$ **do**
1. $u_{k+1} \leftarrow \mathbb{S}^{U_{ad}}_{\beta \cdot s}\left(u_k - s\nabla_{\varepsilon_k} \hat{J}_2(u_k) + \theta(u_k - u_{k-1})\right)$
2. $\varepsilon_{k+1} = \left(\frac{1}{s}(1-\theta) - \frac{L}{2} - c_2\right)\frac{\|u_k - u_{k-1}\|^2}{c(u_b - u_a)}$, where $c > 0$ is defined in Lemma 7.4.4.
3. $\mu_{k+1} = -\alpha u_{k+1} - S'(u_{k+1})^*(S(u_{k+1}) - z)$ (6.3.9)
4. $B_{k+1} = B(u_{k+1}, \mu_{k+1})$
5. $k = k + 1$

end while

Algorithm 9 (Variable truncated inertial proximal (VTIP) method)

Require: β, $\hat{J}_2$, $u_0 = u_{-1}$, U_{ad}, TOL, $\eta > 1$, $L_0 > 0$, ε_0

Initialize: $B_0 = 1$, $k = 0$, choose $\theta \in [0, 1)$ and some small $c_2 > 0$;

while $\|B_k\| > TOL$ **do**

1. Backtracking: Find the smallest nonnegative integer i such that with $\tilde{L} = \eta^i L_{k-1}$

$$\hat{J}_2(\tilde{u}) \leq \hat{J}_2(u_k) + \left\langle \nabla \hat{J}_2(u_k), \tilde{u} - u_k \right\rangle + \frac{\tilde{L}}{2}\|\tilde{u} - u_k\|^2$$

where $\tilde{u} = \mathbb{S}_{\beta \cdot s}^{U_{ad}} \left(u_k - s\nabla_{\varepsilon_k} \hat{J}_2(u_k) + \theta(u_k - u_{k-1}) \right)$, $s < 2(1-\theta)/(\tilde{L} + 2c_2)$,

2. Set $L_k = \tilde{L}$ and $s_k < 2(1-\theta)/(L_k - 2c_2)$.
3. $u_{k+1} = \mathbb{S}_{\beta \cdot s_k}^{U_{ad}} \left(u_k - s_k \nabla_{\varepsilon_k} \hat{J}_2(u_k) + \theta(u_k - u_{k-1}) \right)$
4. $\varepsilon_{k+1} = \left(\frac{1}{s}(1-\theta) - \frac{L}{2} - c_2 \right) \frac{\|u_k - u_{k-1}\|^2}{c(u_b - u_a)}$, where $c > 0$ is defined in Lemma 7.4.4.
5. $\mu_{k+1} = -\alpha u_{k+1} - S'(u_{k+1})^*(S(u_{k+1}) - z)$ (6.3.9)
6. $B_{k+1} = B(u_{k+1}, \mu_{k+1})$
7. $k = k + 1$

end while

It is easily verified that Algorithm 8 and Algorithm 9 are special cases of Algorithm 7, i.e., they fulfill all the requirements of Algorithm 7.

7.2. A special case – The fast truncated proximal scheme (FTP)

Now, we would like to discuss the special case, where $s_k = \frac{1}{L}$ and $\theta_k = 0$ because this will lead to a faster convergence if Assumption 1 is fulfilled as we will see in Subsection 7.3.2.

The following Algorithm implements a proximal scheme

Algorithm 10 (Truncated proximal (TP) method)

Require: β, $\hat{J}_2$, u_0, U_{ad}, TOL, ε_0

Initialize: $v_0 = u_0$; $t_0 = 1$; $B_0 = 1$; $k = 1$

Calculate $L(\hat{J}_2) = \lambda_{max}(\nabla^2 \hat{J}_2)$

while $\|B_{k-1}\| > TOL$ **do**

1. $\varepsilon_k := \frac{\varepsilon_0}{k}$
2. $u_k = \mathbb{S}^{U_{ad}}_{\frac{\beta}{L}} \left(u_{k-1} - \frac{1}{L} \nabla_{\varepsilon_k} \hat{J}_2(u_{k-1}) \right)$
3. $\mu_k = -\alpha u_k - S'(u_k)^*(S(u_k) - z)$ (6.3.9)
4. $B_k = B(u_k, \mu_k)$
5. $k = k + 1$

end while

This scheme is discussed in [BT11] for the case of finite-dimensional optimization problems without the truncation.

In [Nes83], an acceleration strategy for proximal methods applied to convex optimization problems fulfilling (7.0.3) is formulated, that improves the rate of convergence of these schemes from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$. We will see in Subsection 7.3.2 that this also holds for the infinite dimensional truncated version. Specifically, one defines the sequence $\{t_k, v_k\}$ with

$$t_0 = 1, \qquad t_k := 1 + \sqrt{1 + 4t_{k-1}^2}/2, \tag{7.2.1}$$

and

$$v_0 := u_0, \qquad v_k := u_k + \frac{(t_{k-1} - 1)}{t_k}(u_k - u_{k-1}). \tag{7.2.2}$$

Correspondingly, the optimization variable u_k is updated by the following

$$u_k \leftarrow \mathbb{S}^{U_{ad}}_{\frac{\beta}{L}} \left(v_{k-1} - \frac{1}{L} \nabla \hat{J}_2(v_{k-1}) \right).$$

This procedures is summarized in the following algorithm.

Algorithm 11 (Fast truncated proximal (FTP) method)

Require: β, $\hat{J}_2$, u_0, U_{ad}, TOL, ε_0

Initialize: $v_0 = u_0$; $t_0 = 1$; $B_0 = 1$, $k = 1$;

Calculate $L(\hat{J}_2) = \lambda_{max}(\nabla^2 \hat{J}_2)$

while $\|B_{k-1}\| > TOL$ **do**

1. $\varepsilon_k := \frac{\varepsilon_0}{(k+1)^3}$
2. $u_k = \mathbb{S}^{U_{ad}}_{\frac{\beta}{L}} \left(v_{k-1} - \frac{1}{L} \nabla_{\varepsilon_k} \hat{J}_2(v_{k-1}) \right)$
3. $\mu_k = -\alpha u_k - S'(u_k)^*(S(u_k) - z)$ (6.3.9)
4. $B_k = B(u_k, \mu_k)$
5. $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$
6. $v_k = u_k + \left(\frac{t_{k-1}-1}{t_k} \right) (u_k - u_{k-1})$
7. $k = k + 1$

end while

7.3. Convergence analysis of truncated inertial proximal methods

In this section, we investigate the convergence of our truncated proximal schemes. In the following we assume that the error of the truncated gradient has the following upper bound

$$\|\nabla_\varepsilon \hat{J}_2(u) - \nabla \hat{J}_2(u)\| \leq c\varepsilon. \tag{7.3.1}$$

This assumption is discussed in the next section for our elliptic and parabolic optimal control problems separately. We refer to the estimation error of the truncated gradient in step k as follows

$$e_k := \nabla_{\varepsilon_k} \hat{J}_2(u_k) - \nabla \hat{J}_2(u_k), \quad \text{where } \|e_k\| \leq c\varepsilon_k.$$

7.3.1. Convergence of the GTIP method

In this section, we investigate the convergence of our GTIP scheme and therefore also for CTIP and VTIP. Notice that our analysis differs considerably from that presented in [OCBP14] where finite-dimensional problems and exact inversion are considered.

In order to prove the convergence of the GTIP method, we use the strategy of [OCBP14] and extend it to the case of infinite dimensions and non-exact inversion. First, we need the following two lemmas.

Lemma 7.3.1. *Let $v_1, v_2 \in U_{ad}$ and let w be given by*

$$w = \mathbb{S}^{U_{ad}}_{\beta \cdot s_k} \left(v_1 - s_k \hat{J}_2'^{\varepsilon_k}(v_1) + \theta_k (v_1 - v_2) \right).$$

Then, there exists $\gamma(v_1, v_2) \in \partial \hat{J}_1(w)$, such that , for all $u \in U_{ad}$, the following holds

$$\left\langle \gamma(v_1, v_2) + \frac{1}{s_k}(w - v_1) + \nabla \hat{J}_2(v_1) + e_k - \frac{\theta_k}{s_k}(v_1 - v_2), u - w \right\rangle \geq 0. \tag{7.3.2}$$

Proof. Inequality (7.3.2) is the variational inequality that characterizes the solution to the following problem

$$w = \underset{u \in U_{ad}}{\arg\min} \left\{ \hat{J}_1(u) + \frac{1}{2s_k} \left\| u - \left(v_1 - s_k J_2'^{\varepsilon_k}(v_1) + \theta_k(v_1 - v_2) \right) \right\|^2 \right\}. \quad \square$$

The next Lemma is an extension of Proposition 4.7 in [OCBP14]. Therefore, we define

$$H_\delta(u, v) := \hat{J}(u) + \delta \|u - v\|^2 \quad \text{and} \quad \Delta_k := \|u_k - u_{k-1}\|.$$

Lemma 7.3.2.

(a) *We have that*

$$H_{\delta_{k+1}}(u_{k+1}, u_k) \leq H_{\delta_k}(u_k, u_{k-1}) - \gamma_k \Delta_k^2 + c\varepsilon_k(u_b - u_a).$$

(b) *The sequence $(H_{\delta_k}(u_k, u_{k-1}))_k$ is monotonically decreasing and thus converging.*

(c) *It holds that $\sum_{k=0}^{\infty} \Delta_k^2 < \infty$ and therefore $\lim_{k\to\infty} \Delta_k = 0$.*

Proof.

(a) Using inequality (7.0.4) and (7.0.3) with $u = u_{k+1}$ and $v = u_k$, we obtain

$$\begin{aligned} \hat{J}(u_{k+1}) \leq \hat{J}(u_k) &+ \left\langle \nabla \hat{J}_2(u_k), u_{k+1} - u_k \right\rangle \\ &+ \frac{L_k}{2}\|u_{k+1} - u_k\|^2 + \langle \gamma(u_k, u_{k-1}), u_{k+1} - u_k \rangle, \end{aligned}$$

and using (7.3.2) with $u = u_k$, $v_1 = u_k$, $v_2 = u_{k-1}$ and $w = u_{k+1}$ in the above inequality, we have

$$\begin{aligned} \hat{J}(u_{k+1}) \leq \hat{J}(u_k) &- \left(\frac{1}{s_k} - \frac{L_k}{2} \right) \|u_{k+1} - u_k\|^2 \\ &+ \frac{\theta_k}{s_k} \langle u_k - u_{k-1}, u_{k+1} - u_k \rangle - \langle e_k, u_{k+1} - u_k \rangle \\ \leq \hat{J}(u_k) &- \left(\frac{1}{s_k} - \frac{L_k}{2} - \frac{\theta_k}{2s_k} \right) \|u_{k+1} - u_k\|^2 \\ &+ \frac{\theta_k}{2s_k}\|u_k - u_{k-1}\|^2 + c\varepsilon_k(u_b - u_a), \end{aligned}$$

where in the second inequality, we used $2\langle a, b \rangle \leq \|a\|^2 + \|b\|^2$, the Cauchy-Schwarz inequality, and $\|u_{k+1} - u_k\| \leq (u_b - u_a)$. Now, with $\delta_k = \frac{1}{s_k} - \frac{L_n}{2} - \frac{\theta_k}{2s_k}$ and $\gamma_k = \frac{1}{s_k} - \frac{L_n}{2} - \frac{\theta_k}{s_k}$ as in Algorithm 7, we have

$$\hat{J}(u_{k+1}) + \delta_k \Delta_{k+1} \leq \hat{J}(u_k) + \delta_k \Delta_k^2 - \gamma_k \Delta_k^2 + c\varepsilon_k(u_b - u_a).$$

Hence the claim follows, since δ_k is monotonically decreasing.

(b) From (a), we can conclude that the sequence $(H_{\delta_k}(u_k, u_{k-1}))_k$ is monotonically decreasing if $-\gamma_k \Delta_k^2 + c\varepsilon_k(u_b - u_a) \leq 0$ which is fulfilled due to the algorithms requirement $\varepsilon_k \leq (\gamma_k - c_2)\frac{\|u_k - u_{k-1}\|^2}{c(u_b - u_a)}$. Furthermore $(H_{\delta_k}(u_k, u_{k-1}))_k$ is bounded from below by $\hat{J} \geq 0$ and therefore converges.

(c) Summing up the inequality in (a) from $k = 0, \dots, K$ gives

$$\begin{aligned}\sum_{k=0}^{K} \gamma_k \Delta_k^2 &\leq \sum_{k=0}^{K} \Big(H_{\delta_k}(u_k, u_{k+1}) - H_{\delta_{k+1}}(u_{k+1}, u_k) \Big) + c(u_b - u_a) \sum_{k=0}^{K} \varepsilon_k \\ &= \hat{J}(u_0) - H_{\delta_{K+1}}(u_{K+1}, u_K) + (u_b - u_a) \sum_{k=0}^{K} \varepsilon_k \\ &\leq \hat{J}(u_0) + \sum_{k=0}^{K} (\gamma_k - c \cdot c_2)\Delta_k^2.\end{aligned}$$

Since $c, c_2 > 0$ by the algorithm requirements, the claim follows by letting K tend to infinity. □

Now, we can prove the following theorem.

Theorem 7.3.3.

(a) The sequence $(\hat{J}(u_k))_k$ converges.

(b) There exists a weakly convergent subsequence $(u_{k_j})_j$.

(c) If in addition Assumption 1 hold, then any weak limit u^* of $(u_{k_j})_j$ is a critical point of (7.0.1) and $\hat{J}(u^*) \leq \liminf_{j\to\infty} \hat{J}(u_{k_j})$.

Proof.

(a) With the definition of $H_\delta(u, v)$, it holds that

$$H_{-\delta_k}(u_k, u_{k-1}) \leq \hat{J}(u_k) \leq H_{\delta_k}(u_k, u_{k-1}) \tag{7.3.3}$$

and

$$H_{-\delta_k}(u_k, u_{k-1}) = H_{\delta_k}(u_k, u_{k-1}) - 2\delta_k \Delta_k^2$$

So we can use Lemma 7.3.2 (b) and (c) to show that

$$\lim_{k\to\infty} H_{-\delta_k}(u_k, u_{k-1}) = \lim_{k\to\infty} H_{\delta_k}(u_k, u_{k-1}) - 2\delta_k \Delta_k^2 = \lim_{k\to\infty} H_{\delta_k}(u_k, u_{k-1})$$

and with (7.3.3) and the squeeze theorem this yields

$$\lim_{k\to\infty} \hat{J}(u_k) = \lim_{k\to\infty} H_{\delta_k}(u_k, u_{k-1}).$$

(b) Since $H_{\delta_0}(u_0, u_{-1}) = \hat{J}(u_0)$ and $(H_{\delta_k}(u_k, u_{k-1}))_k$ is monotonically decreasing by Lemma 7.3.2 (a) it holds that the sequence $(u_k)_k$ is contained in the level set $\{u \in U_{ad} : 0 \leq \hat{J}(u) \leq \hat{J}(u_0)\}$ and therefore bounded due to the fact that $\hat{J}(u) \to \infty$ as $\|u\| \to \infty$. Now we can use [Bre11, Theorem 3.18] on weakly converging subsequences and the fact that $L^2(\Omega_T)$ is reflexive to state that there exists a weakly converging subsequence $(u_{k_j})_j$.

(c) Let u^* be the weak limit of the sequence u_{k_j}. Then, from $\|u_{k+1} - u_k\| \to 0$ we have that $u_{k_j+1} \rightharpoonup u^*$. From $\partial \hat{J}(u) = \nabla \hat{J}_2(u) + \partial \hat{J}_1(u)$ and Lemma 7.3.1, it follows that

$$\left\langle \nabla \hat{J}_2(u_{k_j+1}) + \gamma_j - \xi_j, u - u_{k_j+1} \right\rangle \geq 0 \quad \forall u \in U_{ad} \tag{7.3.4}$$

where $\gamma_j \in \partial \hat{J}_1(u_{k_j+1})$ and

$$\xi_j := -\frac{1}{s_{k_j}}(u_{k_j+1} - u_{k_j}) - \nabla \hat{J}_2(u_{k_j}) - e_{k_j} + \frac{\theta_{k_j}}{s_{k_j}}(u_{k_j} - u_{k_j-1}) + \nabla \hat{J}_2(u_{k_j+1}).$$

Therefore we have the following

$$\begin{aligned}\|\xi_j\| &\leq \frac{1}{s_{k_j}}\Delta_{k_j+1} + \frac{\theta_{k_j}}{s_{k_j}}\Delta_{k_j} + \|\nabla \hat{J}_2(u_{k_j+1}) - \nabla \hat{J}_2(u_{k_j})\| + \varepsilon_{k_j} \\ &\leq \left(\frac{1}{s_{k_j}} + L\right)\Delta_{k_j+1} + \frac{\theta_{k_j}}{s_{k_j}}\Delta_{k_j} + \varepsilon_{k_j}.\end{aligned}$$

By Lemma 7.3.2 (c) it follows that $\lim_{j\to\infty} \xi_j = 0$. From Assumption 1, we have the convexity of $\hat{J}_2$ and it follows the monotonicity of $\nabla \hat{J}_2$, see, e.g., [Kac60]; The convexity of $\hat{J}_1$ and the monotonicity of $\nabla \hat{J}_2$ together with [KT09, Remark 3(b)] provides the equivalence between inequality (7.3.4) and

$$\left\langle \nabla \hat{J}_2(u) - \xi_j, u - u_{k_j+1} \right\rangle + \hat{J}_1(u) - \hat{J}_1(u_{k_j+1}) \geq 0 \quad \forall u \in U_{ad}. \tag{7.3.5}$$

Now, letting j pass to infinity , we obtain from the lower semicontinuity of $\hat{J}_1$, that

$$\left\langle \nabla \hat{J}_2(u), u - u^* \right\rangle + \hat{J}_1(u) - \hat{J}_1(u^*) \geq 0 \quad \forall u \in U_{ad},$$

that is, due to [KT09, Remark 3(b)], equivalent to

$$\left\langle \nabla \hat{J}_2(u^*) + \gamma, u - u^* \right\rangle \geq 0 \quad \forall u \in U_{ad},$$

where $\gamma \in \partial \hat{J}_1(u^*)$, such that each weak limit u^* of the sequence $(u_{k_j})_j$ is a critical point of (7.0.1). Furthermore, since $\hat{J}$ is convex, we have that $\hat{J}(u^*) \leq \liminf_{j\to\infty} \hat{J}(u_{k_j})$. □

Next, we define the proximal residual and state its convergence rate.

Definition 7.3.1. *The proximal residual is defined by*

$$r(u) := u - \mathbb{S}_{\beta}^{U_{ad}}\left(u - \nabla \hat{J}_2(u)\right)$$

Note that

$$\begin{aligned} r(u) = 0 \Leftrightarrow u = \mathbb{S}_{\beta}^{U_{ad}}(u - \nabla \hat{J}_2(u)) &= \operatorname*{arg\,min}_{v \in U_{ad}} \left\{ \beta \|v\|_{L^1} + \frac{1}{2} \left\| v - \left(u - \nabla \hat{J}_2(u)\right) \right\|^2 \right\} \\ \Leftrightarrow \exists \gamma(u) \in \partial \|u\|_{L^1} &: \left\langle \beta\gamma(u) + \nabla \hat{J}_2(u), \bar{u} - u \right\rangle \geq 0 \quad \text{for all} \quad \bar{u} \in U_{ad} \end{aligned}$$

which is exactly the optimality condition, see (6.3.1). To prove the convergence rate, we need the following two lemmas.

Lemma 7.3.4. *Let $u, v \in L^2(\Omega_T)$, then the function $p : \mathbb{R}^+ \to \mathbb{R}^+$ with*

$$p(s) := \frac{1}{s} \left\| u - \mathbb{S}_{s \cdot \beta}^{U_{ad}}\left(u - s \nabla \hat{J}_2(u)\right) \right\|,$$

is decreasing in s and the function $q : \mathbb{R}^+ \to \mathbb{R}^+$ with

$$q(s) := \left\| u - \mathbb{S}_{s \cdot \beta}^{U_{ad}}\left(u - s \nabla \hat{J}_2(u)\right) \right\|,$$

is increasing in s.

Proof. See [Nes13, Lemma 2]. □

Lemma 7.3.5. *Let $s > 0$, then*

$$\left\| \mathbb{S}_{s \cdot \beta}^{U_{ad}}(u) - \mathbb{S}_{s \cdot \beta}^{U_{ad}}(v) \right\| \leq \|u - v\| \quad \textit{for all} \quad u, v \in L^2(\Omega_T).$$

Proof. See [BC11, Proposition 12.27]. □

The next Lemma gives a relationship between the Δ_k from Lemma 7.3.2 and the proximal residual $r(u_k)$.

Lemma 7.3.6. *Let $(u_k)_k$ be a sequence that is produced from Algorithm 7, then*

$$\sum_{k=0}^{K} \|r(u_k)\| \leq \frac{2}{c_1} \sum_{k=0}^{K} \Delta_{k+1}.$$

Proof. From Lemma 7.3.4, we have

$$1 \leq s \Rightarrow q(1) \leq q(s), \tag{7.3.6}$$

and

$$1 \geq s \Rightarrow p(1) \leq p(s). \tag{7.3.7}$$

Then by using Lemma 7.3.5 and the linearity of $\mathbb{S}$, we obtain

$$\begin{aligned} \theta_k \|u_k - u_{k-1}\| &= \left\| u_k - s\nabla \hat{J}_2(u_k) + \theta_k(u_k - u_{k-1}) - (u_k - s_k \nabla \hat{J}_2(u_k)) \right\| \\ &\geq \left\| u_{k+1} - \mathbb{S}^{U_{ad}}_{s_k\beta}\left(u_k - s_k \nabla \hat{J}_2(u_k)\right) \right\| . \end{aligned} \tag{7.3.8}$$

Now, we can use this to obtain the following inequalities

$$\begin{aligned} \|u_{k+1} - u_k\| &\geq \|u_{k+1} - u_k\| - \theta_k \|u_k - u_{k-1}\| + \left\| u_{k+1} - \mathbb{S}^{U_{ad}}_{s_k\beta}\left(u_k - s_k \nabla \hat{J}_2(u_k)\right) \right\| \\ &\geq \left\| u_k - \mathbb{S}^{U_{ad}}_{s_k\beta}\left(u_k - s_k \nabla \hat{J}_2(u_k)\right) \right\| - \theta_k \|u_k - u_{k-1}\| \\ &\geq \min(1, s_k)\|r(u_k)\| - \|u_k - u_{k-1}\| \\ &\geq c_1 \|r(u_k)\| - \|u_k - u_{k-1}\| \end{aligned}$$

where the first inequality uses (7.3.8), the second uses the triangular equation, and the third arises from (7.3.6), (7.3.7) and $\theta_k < 1$. The claim follows by summing both sides for $k = 0, \dots, K$ and applying $u_{-1} = u_0$. □

Now we can state the desired convergence result.

Theorem 7.3.7. *Let $(u_k)_k$ be the sequence generated by Algorithm 7, then the following holds*

$$\min_{0 \leq k \leq K} \|r(u_k)\|^2 \leq (c_1 c_2)^{-1} \frac{2\hat{J}(u_0)}{K+2}$$

Proof. Summing up the inequality of Lemma 7.3.2 (a), for $k = 0, \dots, K+1$, and applying $u_0 = u_{-1}$ and $\hat{J}(u_{K+1}) \geq 0$ gives

$$\begin{aligned} 0 &\leq \hat{J}(u_0) - \sum_{k=0}^{K+1} \gamma_k \|u_k - u_{k-1}\|^2 + (u_b - u_a) \sum_{k=0}^{K+1} \varepsilon_k \\ &\leq \hat{J}(u_0) - \sum_{k=0}^{K+1} \gamma_k \|u_k - u_{k-1}\|^2 + \sum_{k=0}^{K+1} (\gamma_k - c_2)\|u_k - u_{k-1}\|^2 \\ &\leq \hat{J}(u_0) - \sum_{k=0}^{K+1} c_2 \|u_k - u_{k-1}\|^2 \leq \hat{J}(u_0) - c_2(K+2) \min_{0 \leq k \leq K} \Delta_{k+1}. \end{aligned}$$

By using this and Lemma 7.3.6, we obtain

$$\min_{0 \leq k \leq K} \|r(u_k)\|^2 \leq \frac{2}{c_1} \min_{0 \leq k \leq K} \Delta_{k+1} \leq (c_1 c_2)^{-1} \frac{2\hat{J}(u_0)}{K+2}.$$ □

7.3.2. Fast convergence of the FTP method

In this subsection we investigate the faster convergence of the FTP method. To prove this, we need convexity of the differentiable part $\hat{J}_2$ such that (7.0.4) holds. Therefore we require Assumption 1 to be fulfilled.

First, we define

$$Q_L(u,v) := \beta\|u\|_{L^1} + \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \frac{L}{2}\|u - v\|^2,$$
$$Q_L^e(u,v) := \beta\|u\|_{L^1} + \hat{J}_2(v) + \left\langle \nabla \hat{J}_2(v), u - v \right\rangle + \frac{L}{2}\|u - v\|^2 + \langle e, u - v\rangle\,,$$

and

$$p_L^e(v) := \operatorname*{arg\,min}_{u\in U_{ad}} \{Q_L^e(u,v)\}\,, \tag{7.3.9}$$

such that one step of Algorithm 10, resp. Algorithm 11, can be written as follows

$$u_k = P_L^{e_{k-1}}(u_{k-1}), \quad \text{resp.} \quad u_k = P_L^{e_{k-1}}(v_{k-1}).$$

In order to prove the convergence of the TP method, we need the following two lemmas.

Lemma 7.3.8. *For any $v \in U_{ad}$, one has $w = P_L^e(v)$ iff there exists $\gamma(v) \in \partial\|w\|_{L^1}$, the subdifferential of $\|\cdot\|_{L^1}$, such that*

$$\left\langle \nabla \hat{J}_2(v) + L(w - v) + \beta\gamma(v) + e, u - w \right\rangle \geq 0, \quad \forall u \in U_{ad}. \tag{7.3.10}$$

Proof. This is immediate from the variational inequality of (7.3.9). For a proof see, e.g., [ET99]. □

Lemma 7.3.9. *Let $v \in U_{ad}$ and $L > L(\hat{J}_2)$, then for any $u \in U_{ad}$, we have*

$$\hat{J}(u) - \hat{J}(P_L^e(v)) \geq \frac{L}{2}\|P_L^e(v) - v\|^2 + L\,\langle v - u, P_L^e(v) - v\rangle + \langle P_L^e(v) - u, e\rangle\,.$$

Proof. From (7.0.4), we have

$$\hat{J}(P_L^e(v)) \leq Q_L(P_L^e(v), v),$$

and therefore

$$\hat{J}(u) - \hat{J}(P_L^e(v)) \geq \hat{J}(u) - Q_L(P_L^e(v), v). \tag{7.3.11}$$

Now, since $\beta\|\cdot\|_{L^1}$ and $\hat{J}_2$ are convex, we have

$$\begin{aligned} \beta\|u\|_{L^1} &\geq \beta\|P_L^e(v)\|_{L^1} + \langle u - P_L^e(v), \beta\gamma(v)\rangle \\ \text{and} \quad \hat{J}_2(u) &\geq \hat{J}_2(v) + \left\langle u - v, \nabla \hat{J}_2(v) \right\rangle. \end{aligned}$$

Summing the above inequalities gives

$$\hat{J}(u) \geq \beta\|P_L^e(v)\|_{L^1} + \langle u - P_L^e(v), \beta\gamma(v)\rangle + \hat{J}_2(v) + \left\langle u - v, \nabla \hat{J}_2(v) \right\rangle, \tag{7.3.12}$$

thus using (7.3.10), (7.3.12), and the definition of Q_L in (7.3.11) gives the following

$$\begin{aligned}\hat{J}(u) - \hat{J}(P_L^e(v)) &\geq -\frac{L}{2}\|P_L^e(v) - v\|^2 + \left\langle u - P_L^e(v), \nabla \hat{J}_2(v) + \beta\gamma(v)\right\rangle \\ &\geq -\frac{L}{2}\|P_L^e(v) - v\|^2 + L\left\langle u - P_L^e(v), v - P_L^e(v)\right\rangle + \left\langle P_L^e(v) - u, e\right\rangle \\ &= \frac{L}{2}\|P_L^e(v) - v\|^2 + L\left\langle v - u, P_L^e(v) - v\right\rangle + \left\langle P_L^e(v) - u, e\right\rangle . \qquad \square\end{aligned}$$

Now, we prove an $\mathcal{O}(1/k)$ convergence rate for Algorithm 10 (TP scheme).

Theorem 7.3.10. *Let (u_k) be the sequence generated by Algorithm 10 and u^* be the solution of (6.0.2) with linear or bilinear elliptic equality constraints; let c be determined by (7.4.3) resp. (7.4.5). Then for any $k \geq 1$, we have*

$$\hat{J}(u_k) - \hat{J}(u^*) \leq \frac{L(\hat{J}_2)\|u_0 - u^*\|^2 + 2c\|u_b - u_a\| \cdot \varepsilon_0}{2k}. \tag{7.3.13}$$

Proof. Using Lemma 7.3.9 with $u = u^*$, $v = u_n$ and $L = L(\hat{J}_2)$ we obtain

$$\begin{aligned}\frac{2}{L}(J(u^*) - J(u_{n+1})) &\geq \|u_{n+1} - u_n\|^2 + 2\left\langle u_n - u^*, u_{n+1} - u_n\right\rangle + \frac{2}{L}\left\langle u_{n+1} - u^*, e_k\right\rangle \\ &= \|u^* - u_{n+1}\|^2 - \|u^* - u_n\|^2 + \frac{2}{L}\left\langle u_{n+1} - u^*, e_k\right\rangle\end{aligned}$$

Summing this inequality over $n = 0, \dots, k-1$ gives

$$\begin{aligned}&\frac{2}{L}\left(kJ(u^*) - \sum_{n=0}^{k-1} J(u_{n+1})\right) \\ &\qquad \geq \|u^* - u_k\|^2 + \|u^* - u_0\|^2 + \frac{2}{L}\sum_{n=0}^{k-1}\left\langle u_{n+1}, e_k\right\rangle - \frac{2}{L}k\left\langle u^*, e_k\right\rangle .\end{aligned} \tag{7.3.14}$$

Using Lemma 7.3.9 one more time with $u = v = u_n$, we obtain

$$\frac{2}{L}(J(u_n) - J(u_{n+1})) \geq \|u_n - u_{n+1}\|^2 + \frac{2}{L}\left\langle u_{n+1} - u_n, e_k\right\rangle$$

Multiplying this inequality by n and summing again over $n = 0, \dots, k-1$ gives

$$\begin{aligned}&\frac{2}{L}\sum_{n=0}^{k-1}(nJ(u_n) - (n+1)J(u_{n+1}) + J(u_{n+1})) \\ &\geq \sum_{n=0}^{k-1} n\|u_n - u_{n+1}\|^2 + \frac{2}{L}\sum_{n=0}^{k-1}(-n\left\langle u_n, e_k\right\rangle + (n+1)\left\langle u_{n+1}, e_k\right\rangle - \left\langle u_{n+1}, e_k\right\rangle),\end{aligned}$$

which simplifies to the following

$$\frac{2}{L}\left(-kJ(u_k) + \sum_{n=0}^{k-1} J(u_{n+1})\right) \geq \sum_{n=0}^{k-1} n\|u_n - u_{n+1}\|^2 + \frac{2}{L}k\left\langle u_k, e_k\right\rangle - \frac{2}{L}\sum_{n=0}^{k-1}\left\langle u_{n+1}, e_k\right\rangle . \tag{7.3.15}$$

Adding (7.3.14) and (7.3.15) together, we get

$$\frac{2k}{L}(J(u^*) - J(u_k)) \geq \|u^* - u_k\|^2 + \sum_{n=0}^{k-1} n\|u_n - u_{n+1}\|^2 - \|u^* - u_0\|^2 + \frac{2}{L}k\left\langle u_k - u^*, e_k\right\rangle,$$

and hence with $\varepsilon_k = \frac{\varepsilon_0}{k}$ and $u_a \leq u_k, u^* \leq u_b$ it follows that

$$\begin{aligned} J(u_k) - J(u^*) &\leq \frac{L\|u_0 - u^*\|^2}{2k} + L\left\langle u^* - u_k, e\right\rangle \leq \frac{L\|u_0 - u^*\|^2}{2k} + c\|u^* - u_k\| \cdot \varepsilon_k \\ &\leq \frac{L(\hat{J}_2)\|u_0 - u^*\|^2 + 2c\|u^* - u_k\| \cdot \varepsilon_0}{2k} \\ &\leq \frac{L(\hat{J}_2)\|u_0 - u^*\|^2 + 2c\|u_b - u_a\|\varepsilon_0}{2k}. \end{aligned}$$

□

Next, we present a convergence result for the FTP method. For this purpose, we need the following lemma.

Lemma 7.3.11. *Let (u_k), (v_k) and (t_k) be the sequences generated by Algorithm 11, let e_k be the error of the truncated gradient, and let u^* be the solution to (6.0.2), then for any $k \geq 1$, we have*

$$\frac{2}{L}t_{k-1}^2 w_k - \frac{2}{L}t_k w_{k+1} \geq \|r_{k+1}\|^2 - \|r_k\|^2 + \frac{2}{L}t_k\left\langle r_{k+1}, e_k\right\rangle,$$

with $w_k := J(u_k) - J(u^*)$, $r_k := t_{k-1}u_k - (t_{k-1} - 1)u_{k-1} - u^*$.

Proof. We apply Lemma 7.0.1 at the points $(u := u_k, v := v_k)$ and likewise at the points $(u := u^*, v := v_k)$. We obtain the following

$$\begin{aligned} 2L^{-1}(w_k - w_{k-1}) &\geq \|u_{k+1} - v_k\|^2 + 2\left\langle u_{k+1} - v_k, v_k - u_k\right\rangle + 2L^{-1}\left\langle u_{k+1} - u_k, e_k\right\rangle, \\ -2L^{-1}w_{k-1} &\geq \|u_{k+1} - v_k\|^2 + 2\left\langle u_{k+1} - v_k, v_k - u^*\right\rangle + 2L^{-1}\left\langle u_{k+1} - u^*, e_k\right\rangle, \end{aligned}$$

where we used the fact that $u_{k+1} = p_L^e(v_k)$. Now, we multiply the first inequality above by $(t_k - 1)$ and add it to the second inequality to obtain the following

$$\begin{aligned} &\frac{2}{L}((t_k - 1)w_k - t_k w_{k+1}) \\ &\quad\geq t_k\|u_{k+1} - v_k\|^2 + 2\left\langle u_{k+1} - v_k, t_k v_k - (t_k - 1)u_k - u^*\right\rangle \\ &\qquad + \frac{2}{L}t_k\left\langle u_{k+1} - u_k, e_k\right\rangle + \frac{2}{L}\left\langle u_k - u^*, e_k\right\rangle. \end{aligned}$$

Multiplying this inequality by t_k and using $t_{k-1}^2 = t_k^2 - t_k$, which holds due to (7.2.1), we obtain

$$\begin{aligned} &\frac{2}{L}((t_{k-1}^2 w_k - t_k^2 w_{k+1}) \\ &\quad\geq \|t_k(u_{k+1} - v_k)\|^2 + 2t_k\left\langle u_{k+1} - v_k, t_k v_k - (t_k - 1)u_k - u^*\right\rangle \\ &\qquad + \frac{2}{L}t_k\left\langle t_k u_{k+1} - (t_k - 1)u_k - u^*, e_k\right\rangle. \end{aligned}$$

Applying the Pythagoras relation

$$\|a-b\|^2 + 2\langle b-a, a-c\rangle = \|b-c\|^2 - \|a-c\|^2,$$

to the right-hand side of the last inequality with

$$a := t_k v_k, \quad b := t_k u_{k+1}, \quad c := (t_k - 1)u_k + u^*,$$

we obtain

$$\begin{aligned}\frac{2}{L}((t_{k-1}^2 w_k - t_k^2 w_{k+1}) \\ &\geq \|t_k u_{k+1} - (t_k-1)u_k - u^*\|^2 - \|t_k v_k - (t_k-1)u_k - u^*\|^2 \\ &\quad + \frac{2}{L} t_k \langle t_k u_{k+1} - (t_k-1)u_k - u^*, e_k\rangle .\end{aligned}$$

Therefore, with v_k (see (7.2.2)) and r_k defined as

$$t_k v_k = t_k u_k + (t_{k-1}-1)(u_k - u_{k-1}), \quad r_k := t_{k-1}u_k - (t_{k-1}-1)u_{k-1} - u^*,$$

it follows that

$$\frac{2}{L}t_{k-1}^2 w_k - \frac{2}{L}t_k w_{k+1} \geq \|r_{k+1}\|^2 - \|r_k\|^2 + \frac{2}{L}t_k \langle r_{k+1}, e_k\rangle . \qquad \square$$

We also have the following lemmas.

Lemma 7.3.12. *The positive sequence* (t_k) *generated by the FTP scheme via (7.2.1) with* $t_0 = 1$ *satisfies* $(k+2)/2 \leq t_k \leq k+1$ *for all* $k \geq 0$.

Proof. The proof is immediate by mathematical induction. $\square$

Lemma 7.3.13. *Let* (a_k) *and* (b_k) *be positive sequences of reals and* (c_k) *be a sequence of reals satisfying*

$$a_k + b_k \geq a_{k+1} + b_{k+1} + c_{k+1} \quad \forall k \geq 1 \text{ and } a_1 + b_1 + c_1 \leq d,\ d > 0.$$

Then, $a_k \leq d - \sum_{n=1}^k c_n$.

Proof. The proof is immediate by mathematical induction. $\square$

Now, we can prove a convergence rate of $\mathcal{O}(1/k^2)$ for Algorithm 11 (FTP scheme).

Theorem 7.3.14. *Let* (u_k) *be the sequence generated by Algorithm 11, let* u^* *be the solution to (6.0.2) with linear or bilinear elliptic equality constraints; let* c *be determined by (7.4.3) resp. (7.4.5). Then for any* $k \geq 0$, *the following holds*

$$J(u_k) - J(u^*) \leq \frac{2L(\hat{J}_2)\|u_0 - u^*\|^2 + 2c\|u_b - 2u_a\|\varepsilon_0}{(k+1)^2} \tag{7.3.16}$$

Proof. Let us define the quantities

$$a_k := \frac{2}{L} t_{k-1}^2 w_k, \quad b_k := \|r_k\|^2, \quad c_k := \frac{2}{L} t_{k-1} \langle r_k, e_k \rangle, \quad d := \|u_0 - u^*\|^2.$$

As in Lemma 7.3.11, we define $w_k := J(u_k) - J(u^*)$. Then, by Lemma 7.3.11, the following holds for every $k \geq 1$

$$a_k - a_{k+1} \geq b_{k+1} - b_k + c_{k+1} \Leftrightarrow a_k + b_k \geq a_{k+1} + b_{k+1} + c_{k+1},$$

and hence assuming that $a_1 + b_1 + c_1 \leq d$ holds true, invoking Lemma 7.3.13, we obtain

$$\frac{2}{L} t_{k-1}^2 w_k \leq \|x_0 - x^*\|^2 + \frac{2}{L} t_{k-1} k \max_{n=1,\ldots,k} (\|r_n\|)\|e_n\|,$$

which combined with $t_{k-1} \geq (k+1)/2$ (Lemma 7.3.12) gives the following

$$w_k \leq \frac{2L\|u_0 - u^*\|^2}{(k+1)^2} + 2c \max_{n=1,\ldots,k} (\|r_n\|) \cdot \varepsilon_k. \tag{7.3.17}$$

Furthermore with Lemma 7.3.12 and $u_a \leq u^*, u_k \leq u_b$, we have that

$$\|r_n\| = \|t_{n-1}u_n - (t_{n-1} - 1)u_{n-1} - u^*\| \leq \left\| n u_b - \left(\frac{n-1}{2} u_a + u_a \right) \right\| \leq n \|u_b - 2u_a\|,$$

which combined with (7.3.17) and $\varepsilon_k = \frac{\varepsilon_0}{(k+1)^3}$ gives the following

$$w_k \leq \frac{2L(\hat{J}_2)\|u_0 - u^*\|^2 + 2c\|u_b - 2u_a\|\varepsilon_0}{(k+1)^2}.$$

What remains to be proved is the validity of the relation $a_1 + b_1 + c_1 \leq d$. Since $t_0 = 1$, we have

$$a_1 = \frac{2}{L} t_0^2 w_1 = \frac{2}{L} w_1, \quad b_1 = \|r_1\|^2 = \|u_1 - u^*\|^2, \quad c_1 = 2 \langle u_1 - u^*, e_1 \rangle.$$

Applying Lemma 7.0.1 to the points $u := u^*$ and $v = v_0 = u_0$, we get

$$\begin{aligned} \frac{2}{L}(J(u^*) - J(u_1)) &\geq \|u_1 - v_0\|^2 + 2\langle v_0 - u^*, u_1 - v_0 \rangle + \frac{2}{L} \langle u_1 - u^*, e_1 \rangle \\ &= \|u_1 - u^*\|^2 - \|v_0 - u^*\|^2 + \frac{2}{L} \langle u_1 - u^*, e_1 \rangle, \end{aligned}$$

that is, $-a_1 \geq b_1 - d + c_1 \Leftrightarrow a_1 + b_1 + c_1 \leq d$ holds true. □

Remark 7.3.1. *The TP and FTP methods converge also replacing L with an upper bound of it. In particular, we can prove $\mathcal{O}(1/k^2)$ convergence of the FTP method using a backtracking stepsize rule for the Lipschitz constant (Step 1 in Algorithm 11) as in Algorithm 9.*

We complete this section formulating a fast inexact proximal scheme where the Lipschitz constant L is obtained by forward tracking, (nevertheless we call it backtracking as in [BT11]), thus avoiding any need to compute the reduced Hessian. Our fast truncated proximal backtracking (FTPB) method is presented in Algorithm 12.

Algorithm 12 (Fast truncated proximal backtracking (FTPB) method)

Require: β, $\hat{J}_2$, u_0, U_{ad}, TOL, ε_0, $\eta > 1$, $L_0 > 0$
Initialize: $v_0 = u_0$; $t_0 = 1$; $B_0 = 1$, $k = 1$;
while $\|B_{k-1}\| > TOL$ **do**
1. Backtracking: Find the smallest nonnegative integer i such that with
$\tilde{L} = \eta^i L_{k-1}$

$$\hat{J}_2(\tilde{v}) \leq \hat{J}_2(v_{k-1}) + \left\langle \nabla \hat{J}_2(v_{k-1}), \tilde{v} - v_{k-1} \right\rangle + \frac{\tilde{L}}{2}\|\tilde{v} - v_{k-1}\|^2$$

where $\tilde{v} = \mathbb{S}^{U_{ad}}_{\frac{\beta}{\tilde{L}}} \left(v_{k-1} - \frac{1}{\tilde{L}} \nabla_\varepsilon \hat{J}_2(v_{k-1}) \right)$
2. Set $L_k = \tilde{L}$
3. $\varepsilon_k := \frac{\varepsilon_0}{(k+1)^3}$
1 4. $u_k = \mathbb{S}^{U_{ad}}_{\frac{\beta}{L_k}} \left(v_{k-1} - \frac{1}{L_k} \nabla_\varepsilon \hat{J}_2(v_{k-1}) \right)$
5. $\mu_k = -\alpha u_k - S'(u_k)^*(S(u_k) - z)$ (6.3.9)
6. $B_k = B(u_k, \mu_k)$
7. $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$
8. $v_k = u_k + \left(\frac{t_{k-1}-1}{t_k}\right)(u_k - u_{k-1})$
9. $k = k + 1$
end while

7.4. Proximal methods in optimal control

In this section, we will show that the 'reduced' optimal control problem (6.0.2) fulfills the requirements of the algorithms in the previous section.

First, notice that (6.0.2) has the additive structure (7.0.2)-(7.0.3) where (7.0.2) holds for $\hat{J}_1(u) = \beta\|u\|_{L^1}$, and $\hat{J}_2(u) = \frac{1}{2}\|S(u) - z\|^2 + \frac{\alpha}{2}\|u\|^2$ is at least twice Q-differentiable, it is convex under appropriate conditions discussed in the previous section, and it has Lipschitz-continuous gradient as we prove in the next subsections for the elliptic and parabolic cases separately.

Furthermore, we show that the truncation error of the gradient is bounded from above by inequality (7.3.1).

Now, we discuss the case of elliptic models. First, we prove that the differentiable part of our elliptic optimal control problem has a Lipschitz continuous gradient.

Lemma 7.4.1. *The functional* $\hat{J}_2(u) = \frac{1}{2}\|S(u)-z\|^2 + \frac{\alpha}{2}\|u\|^2$ *has a Lipschitz-continuous gradient for* $S(u) = A^{-1}(f-u)$ *(linear-control case) and for* $S(u) = (A+u)^{-1}f$ *(bilinear-control case).*

Proof. For the linear-control case, we have

$$\begin{aligned}\|\nabla \hat{J}_2(u) - \nabla \hat{J}_2(v)\| &= \|\alpha(u-v) + A^{-*}A^{-1}(u-v)\| \\ &\leq \alpha\|u-v\| + \|A^{-*}A^{-1}\|\|u-v\| \\ &= (\alpha + \|A^{-*}A^{-1}\|_{L^2,L^2})\|u-v\|,\end{aligned}$$

such that we have the Lipschitz-constant $L(\hat{J}_2) = (\alpha + \|A^{-*}A^{-1}\|_{L^2,L^2})$.

For the bilinear-control case, we use the mean value theorem. There exists a $\xi \in U_{ad}$ such that

$$\begin{aligned}\|\nabla \hat{J}_2(u) - \nabla \hat{J}_2(v)\| &\leq \sup_{h\in L^2(\Omega),\|h\|\leq 1} |\nabla \hat{J}_2(u)(h) - \nabla \hat{J}_2(v)(h)| \\ &= \sup_{h\in L^2(\Omega),\|h\|\leq 1} |\hat{J}_2''(\xi)(h, u-v)| \\ &= \sup_{h\in L^2(\Omega),\|h\|\leq 1} |\langle S_b'(\xi)(u-v), S_b'(\xi)(h)\rangle \\ &\qquad + \langle S_b''(\xi)(u-v,h), S_b(\xi)-z\rangle + \alpha\,\langle u-v,,h\rangle| \\ &\leq \left(C_2^2\|f\|^2 + C_1C_3\|f\|^2 - C_3\|f\|\|z\| + \alpha\right)\|u-v\|, \qquad (7.4.1)\end{aligned}$$

for the last inequality, we use (5.1.7),(5.1.14),(5.1.15), that completes the proof. ☐

Now, we investigate the error of the truncated gradient $\nabla_\varepsilon \hat{J}_2(u)$.

Lemma 7.4.2. *The following estimate holds*

$$\left\|\nabla_\varepsilon \hat{J}_2(u) - \nabla \hat{J}_2(u)\right\| \leq c\varepsilon,$$

for some c>0.

Proof. In the linear-control case, we have $\nabla \hat{J}_2(u) = -A^{-*}(A^{-1}(f-u)-z) + \alpha u$. Using (7.0.8) there exist the errors $\tilde{e}_1$, $\tilde{e}_2 \in L^2(\Omega)$ with $\|\tilde{e}_1\|$, $\|\tilde{e}_2\| < \varepsilon$ such that

$$\begin{aligned}\left\|\nabla_\varepsilon \hat{J}_2(u) - \nabla \hat{J}_2(u)\right\| &= \left\|-A^{-*}(A^{-1}(f-u+\tilde{e}_1) - z + \tilde{e}_2) + A^{-*}(A^{-1}(f-u)-z)\right\| \\ &= \left\|-A^{-*}A^{-1}\tilde{e}_1 + A^{-*}\tilde{e}_2\right\| < c\varepsilon, \qquad (7.4.2)\end{aligned}$$

where

$$c = \left\|A^{-*}A^{-1}\right\| + \left\|A^{-*}\right\|. \qquad (7.4.3)$$

In the bilinear-control case, we have $\nabla\hat{J}_2(u) = -(A+u)^{-*}((A+u)^{-1}f - z)(A+u)^{-1}f + \alpha u$. Furthermore, Theorem 5.1.2 implies that the solution $\tilde{y} := (A+u)^{-1}g$ of the equation $A\tilde{y} + u\tilde{y} = g$ has the following property

$$\|(A+u)^{-1}g\| \le C_1\|g\|, \quad \text{for all } g \in L^2(\Omega). \tag{7.4.4}$$

Since A^* also fulfills (5.1.2) with the same θ, we also have

$$\|(A+u)^{-*}g\| = \|(A^*+u)^{-1}g\| \le C_1\|g\|.$$

We have errors $\tilde{e}_1$, $\tilde{e}_2$, $\tilde{e}_3 \in L^2(\Omega)$ with $\|\tilde{e}_1\|$, $\|\tilde{e}_2\|$, $\|\tilde{e}_3\| < \varepsilon$ such that, using (7.0.8) the following holds

$$\begin{aligned}
\left\|\nabla_\varepsilon\hat{J}_2(u) - \nabla\hat{J}_2(u)\right\| &= \Big\|-(A+u)^{-*}\left((A+u)^{-1}(f+\tilde{e}_1) - z + \tilde{e}_2\right)(A+u)^{-1}(f+\tilde{e}_3) \\
&\qquad +(A+u)^{-*}((A+u)^{-1}f - z)(A+u)^{-1}f\Big\| \\
&= \Big\|-(A+u)^{-*}\left((A+u)^{-1}\tilde{e}_1 + \tilde{e}_2\right)(A+u)^{-1}(f+\tilde{e}_3) \\
&\qquad -(A+u)^{-*}\left((A+u)^{-1}f - z\right)(A+u)^{-1}\tilde{e}_3\Big\| \\
&\le \left\|(A+u)^{-*}\left((A+u)^{-1}\tilde{e}_1 + \tilde{e}_2\right)\right\|\left(\left\|y\right\| + \left\|(A+u)^{-1}\tilde{e}_3\right\|\right) \\
&\qquad + \left\|(A+u)^{-*}\left((A+u)^{-1}f - z\right)\right\|\left\|(A+u)^{-1}\tilde{e}_3\right\| \\
&\le C_1\left\|(A+u)^{-1}\tilde{e}_1 + \tilde{e}_2\right\|\left(C_1\|f\| + C_1\|\tilde{e}_3\|\right) \\
&\qquad + C_1\left(\|y\| + \|z\|\right)C_1\|\tilde{e}_3\| \\
&\le C_1^2\Big[C_1\varepsilon + \varepsilon)(\|f\| + \varepsilon) + (C_1\|f\| + \|z\|)\varepsilon\Big] \\
&\le c\varepsilon,
\end{aligned}$$

where

$$c = C_1^2\Big[2C_1\|f\| + \|f\| + C_1 + 1 + \|z\|\Big]. \tag{7.4.5}$$

For the three last inequalities, we use (7.4.4), (5.1.7), and $\varepsilon < 1$. ☐

Next, we discuss the case of parabolic models. We prove that the differentiable part of our parabolic optimal control problem has a Lipschitz continuous gradient.

Lemma 7.4.3. *The functional $\hat{J}_2(u) = \frac{1}{2}\|S(u) - z\|^2 + \frac{\alpha}{2}\|u\|^2$ has a Lipschitz-continuous gradient.*

Proof. For the linear-control case, we have

$$\begin{aligned}
\|\nabla\hat{J}_2(u) - \nabla\hat{J}_2(v)\|_{L^2(\Omega_T)} &= \|\alpha(u-v) + S^*S(u-v)\|_{L^2(\Omega_T)} \\
&\le \alpha\|u-v\|_{L^2(\Omega_T)} + \|S^*S\|_{L^2,L^2}\|u-v\|_{L^2(\Omega_T)} \\
&= (\alpha + \|S^*S\|_{L^2,L^2})\|u-v\|_{L^2(\Omega_T)},
\end{aligned}$$

such that we have the Lipschitz-constant $L(\hat{J}_2) = (\alpha + \|S^*S\|_{L^2,L^2})$.

For the bilinear-control case, we use the mean value theorem. There exists a $\xi \in U_{ad}$ such that

$$\begin{aligned}
\|\nabla\hat{J}_2(u) - \nabla\hat{J}_2(v)\|_{L^2(\Omega_T)} &\leq \sup_{h\in L^2(\Omega),\|h\|\leq 1} |\nabla\hat{J}_2(u)(h) - \nabla\hat{J}_2(v)(h)| \\
&= \sup_{h\in L^2(\Omega),\|h\|\leq 1} |\hat{J}_2''(\xi)(h, u-v)| \\
&= \sup_{h\in L^2(\Omega),\|h\|\leq 1} |\langle S'(\xi)(u-v), S'(\xi)(h)\rangle \\
&\qquad + \langle S''(\xi)(u-v,h), S(\xi) - z\rangle + \alpha\,\langle u-v,,h\rangle| \\
&\leq \Bigg(c_2^2\left(\|y_0\|_{L^2(\Omega)} + \|f\|\right)^2 + c_1c_3\left(\|y_0\|_{L^2(\Omega)} + \|f\|\right)^2 \\
&\qquad + c_3\left(\|y_0\|_{L^2(\Omega)} + \|f\|\right)\|z\| + \alpha\Bigg)\|u-v\|_{L^2(\Omega_T)},
\end{aligned}$$

for the last inequality, we use (5.2.5),(5.1.14),(5.1.15), that completes the proof. □

Furthermore, since $\hat{J}_1$ is convex, the generalized differential is identical to the subdifferential (see Lemma 6.1.1) and we have

$$\partial^*\hat{J}_1(u) = \partial\hat{J}_1(u) = \{\gamma \in L^2(u) : \langle\gamma, v-u\rangle \leq \hat{J}_1(v) - \hat{J}_1(u) \ \text{ for all } \ v \in U_{ad}\} \quad (7.4.6)$$

Now, we investigate the error of the truncated gradient $\nabla_\varepsilon\hat{J}_2(u)$ for the parabolic model.

Lemma 7.4.4. *The following estimate holds*

$$\left\|\nabla_\varepsilon\hat{J}_2(u) - \nabla\hat{J}_2(u)\right\| \leq c\cdot\varepsilon,$$

for some c>0.

Proof. We start considering the case of bilinear control. Since y^ε satisfies

$$\partial_t y^\varepsilon + Ay^\varepsilon + uy^\varepsilon = f + e_1, \quad y^\varepsilon(\cdot,t) = y_0,$$

for some $e_1 \in L^2(\Omega_T)$ and $\|e_1\| < \varepsilon$, we have that $\tilde{y} := y^\varepsilon - y$ satisfies the following

$$\partial_t\tilde{y} + A\tilde{y} + u\tilde{y} = e_1, \quad \tilde{y}(\cdot,0) = 0,$$

and therefore, using Theorem 5.1.2

$$\|y^\varepsilon - y\| \leq C_1\|e_1\| \leq C_1\varepsilon. \quad (7.4.7)$$

Furthermore, p^ε satisfies

$$-\partial_t p^\varepsilon + A^*p^\varepsilon + up^\varepsilon = z - y^\varepsilon + e_2, \quad \tilde{p}(\cdot,T) = 0,$$

for some $e_2 \in L^2(\Omega_T)$ and $\|e_2\| < \varepsilon$ and since A^* also fulfills (5.1.2) , we can use Theorem 5.1.2 and obtain

$$\|p^\varepsilon\| \leq C_2(\|z\| + \|f\| + \|y_0\|_{L^2(\Omega)} + 2\varepsilon).$$

In addition, we have that $\tilde{p} := p^\varepsilon - p$ satisfies

$$-\partial_t \tilde{p} + A^* \tilde{p} + u\tilde{p} = y - y^\varepsilon + e_2, \quad \tilde{p}(\cdot, T) = 0,$$

such that

$$\|p^\varepsilon - p\| \leq C_3 \|(y - y^\varepsilon) + e_2\| \leq C_3(C_1\varepsilon + \varepsilon) \leq C_4\varepsilon. \tag{7.4.8}$$

Now, we can estimate the error of the gradient $\nabla \hat{J}_2(u) = py + \alpha u$.

$$\begin{aligned}\|\nabla_\varepsilon \hat{J}_2(u) - \nabla \hat{J}_2(u)\| = \|p^\varepsilon y^\varepsilon - py\| = \|p^\varepsilon(y^\varepsilon - y) + y(p^\varepsilon - p)\| &\leq \|p^\varepsilon\| C_4\varepsilon + \|y\| C_1\varepsilon \\ &\leq \Big(C_4C_2(\|f\| + \|y_0\|_{L^2(\Omega)} + \|z\| + 2) + C_1C_5\big(\|f\| + \|y_0\|_{L^2(\Omega)}\big)\Big)\varepsilon \leq c\varepsilon\end{aligned}$$

with $c = C_4C_2(\|f\| + \|y_0\|_{L^2(\Omega)} + \|z\| + 2) + C_1C_5\big(\|f\| + \|y_0\|_{L^2(\Omega)}\big)$.

Now, we prove the inequality for the linear-control case. For the unique solution to

$$\partial_t y + Ay = g, \quad y(\cdot, 0) = y_0,$$

with $g \in L^2(\Omega_T)$, we have the following estimate; see, e.g., [Eva10, Chapter 7];

$$\|y\|_{L^2(0,T;H_0^1(\Omega))} \leq C(\|g\| + \|y_0\|_{L^2(\Omega)}).$$

Since $\tilde{y} := y^\varepsilon - y$ satisfies

$$\partial_t \tilde{y} + A\tilde{y} = e_1, \quad y(\cdot, 0) = 0,$$

for some $e_1 \in L^2(\Omega_T)$ and $\|e_1\| < \varepsilon$, we have that

$$\|y^\varepsilon - y\| \leq C_1 e_1 \leq C_1\varepsilon.$$

Furthermore, $\tilde{p} = p^\varepsilon - p$ satisfies

$$-\partial_t \tilde{p} + A^* \tilde{p} = y - y^\varepsilon + e_2, \quad \tilde{p}(\cdot, T) = 0,$$

for some $e_2 \in L^2(\Omega_T)$ and $\|e_2\| < \varepsilon$, such that

$$\|p^\varepsilon - p\| \leq C_2 \|(y - y^\varepsilon) + e_2\| \leq C_2(C_1\varepsilon + \varepsilon) \leq C_3\varepsilon. \tag{7.4.9}$$

Now, we can estimate the error of the gradient $\nabla \hat{J}_2(u) = p + \alpha u$.

$$\|\nabla_\varepsilon \hat{J}_2(u) - \nabla \hat{J}_2(u)\| = \|p^\varepsilon - p\| \leq C_3\varepsilon$$

which finishes the proof. □

We see, that both our elliptic and parabolic optimal control problem fulfill the assumptions needed for convergence of the proximal methods.

8. Inexact semismooth Newton methods in function space

8.1. The semismooth Newton method

We consider the semismooth Newton method as a benchmark scheme for solving elliptic and parabolic non-smooth optimal control problems.The inexact semismooth Newton (ISSN) method was presented in [MQ95] for finite-dimensional problems and in [Ulb11] for infinite-dimensional problems. In this section, we discuss the ISSN method for infinite-dimensional optimization problems and use it for comparison with our truncated proximal schemes. In this section, to support our use of the ISSN scheme to solve bilinear control problems, we extend two theoretical results in [Ulb11, Sta09].

Now, we discuss the solution of the following nonlinear equation

$$\mathcal{F}(x) = 0.$$

We have the following theorem.

Theorem 8.1.1. *[HIK02, Theorem 1.1] Suppose that x^* is a solution to $\mathcal{F}(x) = 0$ and that $\mathcal{F}$ is generalized differentiable in an open neighborhood U containing x^* with a generalized derivative $\mathcal{G}$. If $\mathcal{G}(x)$ is invertible for all $x \in U$ and $\{\|\mathcal{G}(x)^{-1}\|_{Y,X} : x \in U\}$ is bounded, then the semismooth Newton (SSN) iteration*

$$x^{k+1} = x^k - \mathcal{G}(x^k)^{-1}\mathcal{F}(x^k)$$

converges superlinearly to x^*, provided that $\|x^0 - x^*\|$ is sufficiently small.

An inexact version of the SSN scheme discussed in this theorem is formulated in [Ulb11, Algorithm 3.19], where the direction update d_k to x_k is obtained as follows. Choose a boundedly invertible operator $B_k \in \mathcal{L}(X, Y)$ and compute

$$d_k = -B_k^{-1}\mathcal{F}(x_k). \tag{8.1.1}$$

For this scheme, superlinear convergence is proven in [Ulb11, Theorem 3.20], provided that there exists a $\mathcal{G}_k \in \partial^*\mathcal{F}(x_k)$ such that

$$\lim_{\|d_k\|_X \to 0} \frac{\|(B_k - \mathcal{G}_k)d_k\|_Y}{\|d_k\|_X} = 0.$$

However, this procedure is difficult to realize in practice. For this reason, in our ISSN scheme, the ‘exact’ update step $x^{k+1} = x^k + d_k$ with $d_k = -\mathcal{G}(x^k)^{-1}\mathcal{F}(x^k)$, as discussed in [HIK02], is replaced by $x^{k+1} = x^k + d_k$ with d_k satisfying the following inequality

$$\|\mathcal{G}(x_k)d_k + \mathcal{F}(x_k)\|_Y \leq \eta_k \|\mathcal{F}(x_k)\|_Y. \tag{8.1.2}$$

Our ISSN scheme is given in algorithmic form in Algorithm 13.

Algorithm 13 (Inexact semismooth Newton (ISSN) method)

Require: $\mathcal{F}$, $x_0 \in D$

Initialize: $k = 0$;

while $\mathcal{F}(x_k) = 0$ **do**

1. Calculate the direction d_k such that

$$\|\mathcal{G}(x_k)d_k + \mathcal{F}(x_k)\|_Y \leq \eta_k \|\mathcal{F}(x_k)\|_Y \tag{8.1.3}$$

with $\eta_k < 1$ and $\eta_k \to 0$

2. $x_{k+1} = x_k + d_k$
3. $k = k + 1$

end while

8.2. Convergence of the ISSN scheme

On the basis of the proof of Theorem 3.20 in [Ulb11], we prove the following theorem that states convergence of Algorithm 13. We have

Theorem 8.2.1. *Suppose that x^* is a solution to $\mathcal{F}(x) = 0$ and that $\mathcal{F}$ is generalized differentiable and Lipschitz continuous in an open neighborhood U containing x^* with a generalized derivative $\mathcal{G}$. If $\mathcal{G}(x)$ is invertible for all $x \in U$ and $\{\|\mathcal{G}(x)^{-1}\|_{Y,X} : x \in U\}$ is bounded, then Algorithm 13 converges superlinearly to x^*, provided that $\|x_0 - x^*\|_X$ is sufficiently small.*

Proof. Let $r_k := \mathcal{G}(x_k)d_k + \mathcal{F}(x_k)$ and $v_k := x_k - x^*$. Furthermore, let $\delta > 0$ be so small that $\|x_0 - x^*\|_X < \delta$ and $\mathcal{F}$ is Lipschitz continuous in $x^* + \delta B_X \subset U$ with $L > 0$. Now, we show inductively that $\|x_{k+1} - x^*\| < \delta$ for all k. We assume that $\|x_k - x^*\| < \delta$ for some $k \geq 0$. Then there holds

$$\|\mathcal{F}(x_k)\|_Y \leq L\|v_k\|_X.$$

We estimate the Y-norm of r_k:

$$\|r_k\|_Y \leq \eta_k \|\mathcal{F}(x_k)\|_Y \leq L\eta_k \|v_k\|_X, \tag{8.2.1}$$

Next, using $\mathcal{F}(x^*) = 0$ we obtain

$$\begin{aligned} \mathcal{G}(x_k)v_{k+1} &= \mathcal{G}(x_k)(d_k + v_k) = r_k - \mathcal{F}(x_k) + \mathcal{G}(x_k)v_k \\ &= r_k - [\mathcal{F}(x^* + v_k) - \mathcal{F}(x^*) - \mathcal{G}(x^* + v_k)v_k]. \end{aligned} \tag{8.2.2}$$

This result, the generalized differentiability of $\mathcal{F}$ at x^*, and (8.2.1), give the following

$$\|\mathcal{G}(x_k)v_{k+1}\|_Y = o(\|v_k\|_X) \text{ as } \|v_k\|_Y \to 0. \tag{8.2.3}$$

Hence, for sufficiently small $\delta > 0$, we have

$$\|\mathcal{G}(x_k)v_{k+1}\|_Y \leq \frac{1}{2C_{\mathcal{G}^{-1}}}\|v_k\|_X,$$

with $C_{\mathcal{G}^{-1}} = \sup\{\|\mathcal{G}(x)^{-1}\|_{Y,X} : x \in U\}$ and thus

$$\|v_{k+1}\|_X \leq \|\mathcal{G}(x_k)^{-1}\|_{Y,X}\|\mathcal{G}(x_k)v_{k+1}\|_Y \leq \frac{1}{2}\|v_k\|_X.$$

This gives

$$x_{k+1} \in x^* + \frac{\|v_k\|_X}{2}\bar{B}_X \subset x^* + \frac{\delta}{2}B_X \subset U,$$

which inductively gives $x_k \to x^*$ in Y. Now, we conclude from (8.2.3) that

$$\|v_{k+1}\|_X \leq C_{\mathcal{G}^{-1}}\|\mathcal{G}(x_k)v_{k+1}\|_Y = o(\|v_k\|_X),$$

which completes the proof. □

8.3. Semismooth Newton methods in optimal control

Our purpose is to solve the nonlinear and nonsmooth system (6.3.9)-(6.3.10) by the semismooth Newton iteration. We introduce the operator

$$\mathcal{T} : L^2(\Omega_T) \to L^s(\Omega_T), \quad \mathcal{T}(u) := \mathcal{I}(S'(u)^*(z - S(u))),$$

where $\mathcal{I}$ is the Sobolev embedding (see [Ada75, Theorem 5.4]) of $H_0^1(\Omega)$ into $L^s(\Omega)$ for the elliptic case, resp. $H^1(\Omega_T) \supset H^{2,1}(\Omega_T)$ into $L^s(\Omega_T)$ for the parabolic case, with $s > 2$. This embedding is necessary to show that the function $\mathcal{F}$ defined in (8.3.2) is generalized differentiable.

Now, by using $\bar{\mu} = -\alpha\bar{u} + \mathcal{T}(\bar{u})$ from (6.3.9) and choosing $c := \alpha^{-1}$, equation (6.3.10) becomes to

$$\mathcal{F}(\bar{u}) = 0, \tag{8.3.1}$$

where

$$\begin{aligned}\mathcal{F}(u) := u - \alpha^{-1}\max\{0, \mathcal{T}(u) - \beta\} - \alpha^{-1}\min\{0, \mathcal{T}(u) + \beta\} \\ + \alpha^{-1}\max\{0, \mathcal{T}(u) - \beta - \alpha u_b\} + \alpha^{-1}\min\{0, \mathcal{T}(u) + \beta - \alpha u_a\}.\end{aligned} \tag{8.3.2}$$

The function $\mathcal{F}$ is generalized differentiable (see [Sta09, Theorem 4.2] for the elliptic linear case, analogue for the parabolic and the bilinear case) and a generalized derivative is given by

$$\mathcal{G}(u)(v) = v - \alpha^{-1}\chi_{(\mathcal{I}_- \cup \mathcal{I}_+)}(\mathcal{T}'(u)(v)), \tag{8.3.3}$$

where

$$\begin{aligned}\mathcal{I}_- &:= \{x \in \Omega_T :\ \alpha u_a \leq \mathcal{T}(u) + \beta \leq 0 \ \text{ a.e. in } \Omega_T\} \\ \mathcal{I}_+ &:= \{x \in \Omega_T :\ 0 \leq \mathcal{T}(u) - \beta \leq \alpha u_b \ \text{ a.e. in } \Omega_T\}.\end{aligned}$$

Using Theorem 8.2.1, we can prove the following theorem that guarantees the superlinear convergence of the semismooth Newton method applied to our problems. To prove this, we extend the proof of Theorem 4.3 in [Sta09].

Theorem 8.3.1. *If*

$$C''(u)\|S(u) - z\| < \alpha, \tag{8.3.4}$$

with $C''(u) := \sup_{\|v\|\leq 1} \|S''(u)(v,v)\|$, *then* $\mathcal{G}(u)$ *is invertible for all* $u \in U_{ad}$ *and* $\{\|\mathcal{G}(u)^{-1}\|_{L^2,L^2} : u \in U_{ad}\}$ *is bounded.*

Proof. We denote $V = \Omega$ for the elliptic case and $V = \Omega_T$ for the parabolic case. Furthermore, we define $J := \mathcal{I}_- \cup \mathcal{I}_+$, and for $\mathcal{D} \subset V$ and $v \in L^2(\Omega_T)$ the restriction operator $E_\mathcal{D} : L^2(V) \to L^2(\mathcal{D})$ by $E_\mathcal{D}(v) := v|_\mathcal{D}$. The corresponding adjoint operator is the extension-by-zero operator $E^*_\mathcal{D} : L^2(\mathcal{D}) \to L^2(V)$. We assume that $\mathcal{G}(u)(v) = w$. From (8.3.3), we obtain that $E_{V\setminus J}v = E_{V\setminus J}w$. Thus, $v_J := E_J v \in L^2(J)$ satisfies

$$v_J - \alpha^{-1}E_J\mathcal{T}'(u)(E^*_J v_J) = E_J w + \alpha^{-1}E_J\mathcal{T}'(u)(E^*_{V\setminus J}E_{V\setminus J}w). \tag{8.3.5}$$

Now, we define

$$g(\varphi) := \left\langle E_J w + \alpha^{-1}E_J\mathcal{T}'(u)(E^*_{V\setminus J}E_{V\setminus J}w), \varphi\right\rangle_{L^2(J)},$$

and

$$\begin{aligned}a(v_1, v_2) := \langle v_1, v_2\rangle_{L^2(J)} + \alpha^{-1}\Big[&\langle S(u) - z, S''(u)(E^*_J v_1, E^*_J v_2)\rangle_{L^2(V)} \\ &+ \langle S'(u)(E^*_J v_1), S'(u)(E^*_J v_2)\rangle_{L^2(V)}\Big],\end{aligned}$$

for $\varphi, v_1, v_2 \in L^2(J)$. We use

$$\left\langle \mathcal{T}'(u)(w_1), w_2\right\rangle_{L^2(V)} = \left\langle z - S(u), S''(u)(w_1, w_2)\right\rangle_{L^2(V)} - \left\langle S'(u)(w_1), S'(u)(w_2)\right\rangle_{L^2(V)},$$

to see that (8.3.5) is equivalent to

$$a(v_J, \varphi) = g(\varphi), \quad \text{for all } \varphi \in L^2(J). \tag{8.3.6}$$

Using $\left\langle v_1, v_2 \right\rangle_{L^2(J)} = \left\langle E_J^* v_1, E_J^* v_2 \right\rangle_{L^2(\Omega_T)}$ and $S''(u)(h_1, h_2) = 0$ in the linear case resp. (8.3.4) in the bilinear case we have coercivity of a for $u \in U_{ad}$ and therefore the Lax-Milgram-Lemma can be applied to show that (8.3.5) admits a unique solution $v_J \in L^2(J)$. Moreover, this solution satisfies

$$\|v_J\|_{L^2(J)} \leq \tilde{C} \|g\|_{L^2(J)} \leq C \|w\|_{L^2(V)},$$

with a constant C independent of u. For the last inequality we use the fact that $\mathcal{T}'(u)$ is bounded due to the boundedness of $S(u)$, $S'(u)$ and $S''(u)$ as shown in (5.1.7),(5.1.14), (5.1.15) for the elliptic case resp. (5.2.5),(5.2.9), and (5.2.10) for the parabolic case. □

Remark 8.3.1. *The assumption of Theorem 8.3.1 is equivalent to Assumption 1.*

9. Numerical experiments

In this section, we present results of numerical experiments to validate the computational performance of our FTP method and to demonstrate the convergence rate of $\mathcal{O}(1/k^2)$ proved in Theorem 7.3.14. For benchmarking purposes, the FTP scheme is compared to an inexact semismooth Newton method. Results of numerical experiments demonstrate the computational effectiveness of truncated proximal schemes and successfully validate the theoretical estimates.

9.1. Elliptic models

We start our discussion with the elliptic models. For validation purposes, we formulate control problems for which we know the exact solution. We have

Procedure 1. *(Linear case)*

1. *Choose* $\hat{y} \in H_0^1(\Omega)$ *and* $\hat{p} \in H_0^1(\Omega)$ *arbitrary*
2. *Set* $\hat{u} := \begin{cases} \max\{\frac{-\hat{p}+\beta}{\alpha}, u_a\} & \text{on } \{x \in \Omega : \hat{p}(x) > \beta\} \\ \min\{\frac{-\hat{p}-\beta}{\alpha}, u_b\} & \text{on } \{x \in \Omega : \hat{p}(x) < -\beta\} \\ 0 & \text{elsewhere} \end{cases}.$
3. $\mu := -\hat{p} - \alpha\hat{u}$
4. $f := A\hat{y} + \hat{u}$
5. $z := A^*\hat{p} + \hat{y}$

Lemma 9.1.1. *Procedure 1 provides a solution* $(\hat{y}, \hat{u})$ *of the optimal control problem (6.0.2) with the elliptic model and linear control mechanism.*

Proof. We show that the optimality conditions (6.3.12)–(6.3.15) in Theorem 6.3.2 are fulfilled. In fact, (6.3.12)–(6.3.14) are obviously fulfilled because of 3.– 5. in Procedure 1. Now, we consider different cases to show (6.3.15):

- $|\hat{p}| \leq \beta$: From 2. we have $\hat{u} = 0$ and from 3. $\mu = -\hat{p}$ and therefore

$$\begin{aligned} B(\hat{u}, \mu) = 0 &- \max\{0, c(-\hat{p} - \beta)\} - \min\{0, c(-\hat{p} + \beta)\} \\ &+ \max\{0, -u_b + c(-\hat{p} - \beta)\} + \min\{0, -u_a + c(-\hat{p} + \beta)\} = 0. \end{aligned}$$

- $\hat{p} > \beta$:
  - $\star$ $u_a \le \frac{-\hat{p}+\beta}{\alpha} \le u_b$: From 2. we have $\hat{u} = \frac{-\hat{p}+\beta}{\alpha} < 0$ and from 3. we have $\mu = -\hat{p} - \alpha\hat{u} = -\beta$, therefore

$$\begin{aligned} B(\hat{u},\mu) &= \hat{u} - \max\{0, \hat{u} - c(2\beta)\} - \min\{0, \hat{u})\} \\ &\qquad + \max\{0, \hat{u} - u_b - c(2\beta)\} + \min\{0, \hat{u} - u_a\} \\ &= \hat{u} - 0 - \hat{u} + 0 + 0 = 0. \end{aligned}$$

  - $\star$ $\frac{-\hat{p}+\beta}{\alpha} \le u_a$: From 2. we have $\hat{u} = u_a$ and from 3. we have $\mu = -\hat{p} - \alpha u_a$, therefore

$$\begin{aligned} B(\hat{u},\mu) &= u_a - \max\{0, u_a + c(-\hat{p} - \alpha u_a - \beta)\} - \min\{0, u_a + c(-\hat{p} - \alpha u_a + \beta)\} \\ &\qquad + \max\{0, u_a - u_b + c(-\hat{p} - \alpha u_a - \beta)\} \\ &\qquad + \min\{0, u_a - u_a + c(-\hat{p} - \alpha u_a + \beta)\} \\ &= u_a - 0 - (u_a + c(-\hat{p} - \alpha u_a + \beta)) + 0 + c(-\hat{p} - \alpha u_b + \beta) = 0. \end{aligned}$$

- $\hat{p} < -\beta$
  - $\star$ $u_a \le \frac{-\hat{p}-\beta}{\alpha} \le u_b$: From 2. we have $\hat{u} = \frac{-\hat{p}-\beta}{\alpha} > 0$ and from 3. we have $\mu = -\hat{p} - \alpha\hat{u} = \beta$. So

$$\begin{aligned} B(\hat{u},\mu) &= \hat{u} - \max\{0, \hat{u}\} - \min\{0, \hat{u} + c(2\beta)\} \\ &\qquad + \max\{0, \hat{u} - u_b\} + \min\{0, \hat{u} - u_a + c(2\beta)\} \\ &= \hat{u} - \hat{u} - 0 + 0 + 0 = 0. \end{aligned}$$

  - $\star$ $\frac{-\hat{p}-\beta}{\alpha} \ge u_b$: From 2. we have $\hat{u} = u_b$ and from 3. we have $\mu = -\hat{p} - \alpha u_b$, therefore

$$\begin{aligned} B(\hat{u},\mu) &= u_b - \max\{0, u_b + c(-\hat{p} - \alpha u_b - \beta)\} - \min\{0, u_b + c(-\hat{p} - \alpha u_b + \beta)\} \\ &\qquad + \max\{0, u_b - u_b + c(-\hat{p} - \alpha u_b - \beta)\} \\ &\qquad + \min\{0, u_b - u_a + c(-\hat{p} - \alpha u_b + \beta)\} \\ &= u_b - (u_b + c(-\hat{p} - \alpha u_b - \beta)) - 0 + c(-\hat{p} - \alpha u_b - \beta) + 0 = 0. \end{aligned}$$ □

Procedure 2. *(Bilinear case)*

1. *Choose* $\hat{y} \in H_0^1(\Omega)$ *and* $\hat{p} \in H_0^1(\Omega)$ *arbitrary*
2. *Set* $\hat{u} := \begin{cases} \max\{\frac{-\hat{p}\hat{y}+\beta}{\alpha}, u_a\} & \text{on } \{x \in \Omega : \hat{p}(x)\hat{y}(x) > \beta\} \\ \min\{\frac{-\hat{p}\hat{y}-\beta}{\alpha}, u_b\} & \text{on } \{x \in \Omega : \hat{p}(x)\hat{y}(x) < -\beta\} \\ 0 & \text{elsewhere} \end{cases}.$

3. $\mu := -\hat{p}\hat{y} - \alpha\hat{u}$

4. $f := A\hat{y} + \hat{u}\hat{y}$

5. $z := A^*\hat{p} + \hat{y} + \hat{u}\hat{p}$

Lemma 9.1.2. *Procedure 2 provides a solution $(\hat{y}, \hat{u})$ to the optimal control problem (6.0.2) with the elliptic model and bilinear control mechanism.*

Proof. The proof is similar to the one of the linear case. □

Next, we specify the elliptic operator, the domain of computation, the choice of $\hat{y}$ and $\hat{p}$, and some optimization and numerical parameters. We consider the following examples.

Case 1. *(1 dimensional) $\Omega = (0,1)$, $A = -\Delta$, $u_a \equiv -1$, $\alpha = 0.05$, $\hat{y} = \sin(\pi x)$ and $\hat{p} = 2\beta \sin(2\pi x)$. We discretize Ω with gridsize $h = 1024$. A is discretized by second-order finite differences. Then we have $c_\Omega = \frac{1}{2}$, $a_0 = 0$ and $\theta = 1$ such that (5.1.6) holds. The results are shown in Table 9.1.*

Case 2. *(2 dimensional) $\Omega = (0,1)^2$, $A = -\Delta$, $u_a \equiv -1$, $\alpha = 0.05$, $\hat{y} = \sin(\pi x_1)\sin(\pi x_1)$ and $\hat{p} = 4\beta \sin(2\pi x_1)\sin(\pi x_2)$. We discretize Ω with gridsize $h = 1/256$. A is discretized by second-order finite differences. Then we have $c_\Omega = \frac{1}{4}$, $a_0 = 0$ and $\theta = 1$ such that (5.1.6) holds. The results are shown in Table 9.2.*

We compare the FTP, FTPB and ISSN schemes in terms of computational time. In the FTP method, we calculate the smallest Lipschitz constant as the dominant eigenvalue of $\nabla^2 \hat{J}_2(u)$ with a power iteration. The power iteration is defined by the following scheme.

$$b_{k+1} = \frac{\nabla^2 \hat{J}_2(u) b_k}{\|\nabla^2 \hat{J}_2(u) b_k\|}.$$

. This power iteration is stopped if the difference between two iterates of the norm $\|\nabla^2 \hat{J}_2\|_{L^2,L^2}$ is less or equal than a tolerance of 10^{-5}. For the FTPB method, we use backtracking with $\eta = 1.5$ and $L_0 = 0.001$. All algorithms are stopped if $B(u_k, \mu_k) < 10^{-6}$. We can see in Table 9.1 and 9.2 that the computational performance of the FTP and FTPB methods is comparable to that of the ISSN method.

In order to validate the theoretical rate of convergence of $\mathcal{O}(1/k^2)$, the theoretical upper bound of Theorem 7.3.14 and the actual error of the functional in correspondence to Case 1 and Case 2 with $\beta = 0.1$ and $\alpha = 0.005$, are plotted in Figure 9.1. We see that the observed convergence rate may be faster than the theoretical prediction.

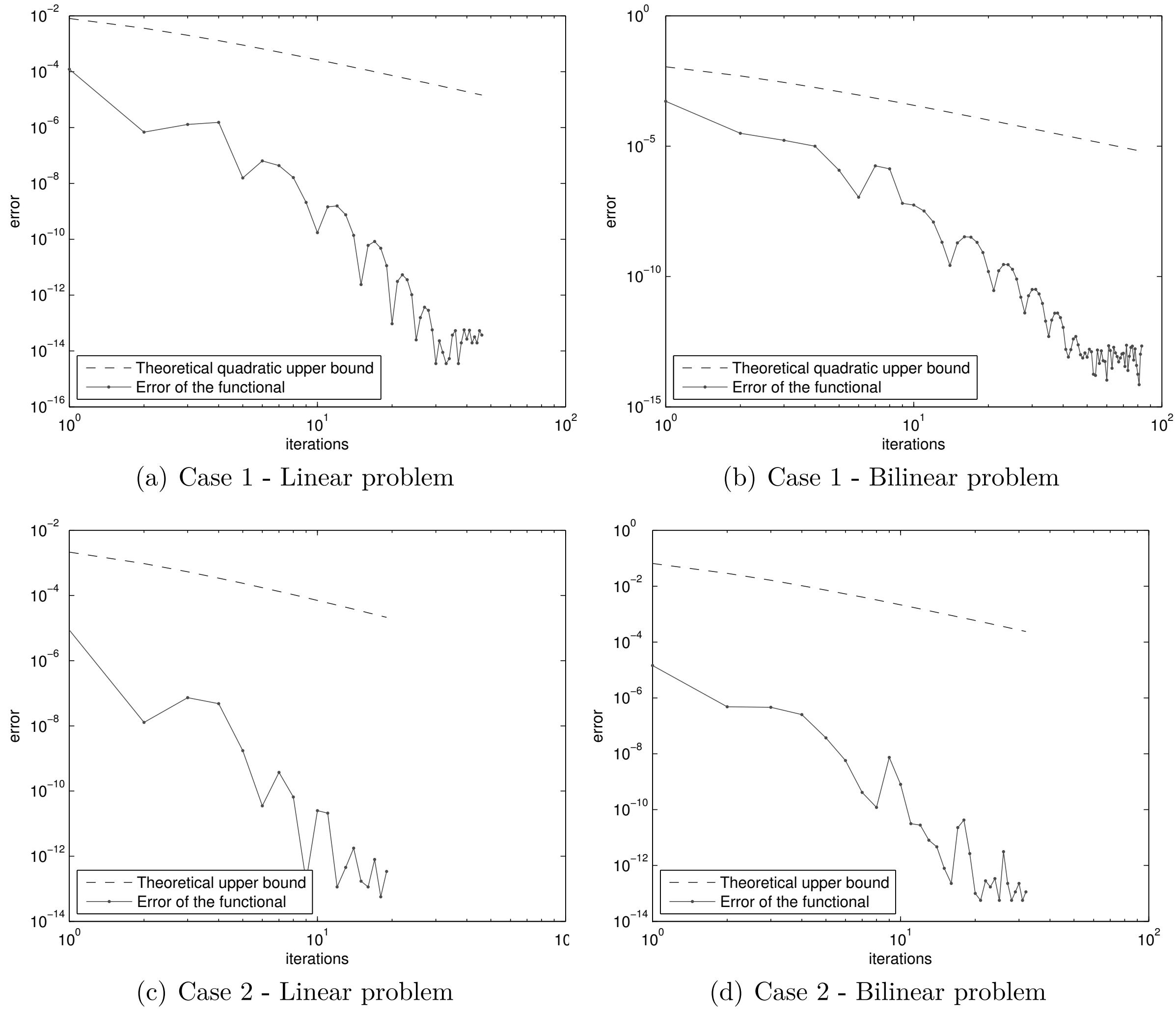
We conclude this section considering a challenging linear- and a bilinear-control case. However the exact solutions are not known. In these cases the target function is not attainable. We have

		linear case ($u_b \equiv 15\beta$)			bilinear case ($u_b \equiv 7\beta$)		
α	β	FTP	FTPB	ISSN	FTP	FTPB	ISSN
0.5	0.1	0.441s	3.86s	0.591s	2.89s	8.62s	4.11s
	0.01	0.333s	8.26s	0.587s	2.07s	9.57s	2.75s
0.05	0.1	2.33s	8.74s	2.56s	6.94s	17.8s	6.62s
	0.01	1.82s	7.78s	1.26s	3.11s	19.42s	4.37s
0.005	0.1	6.48s	51.7s	2.49s	15.0s	7.2s	7.9s
	0.01	6.48s	5.50s	2.68s	8.04s	7.15s	6.61s

Table 9.1.: Case 1 – Comparison of the FTP, FTPB and ISSN methods.

		linear case ($u_b \equiv 15\beta$)			bilinear case ($u_b \equiv 7\beta$)		
α	β	FTP	FTPB	ISSN	FTP	FTPB	ISSN
0.5	0.1	6.55s	34.0s	6.83s	58.3s	156s	123s
	0.01	5.27s	28.6s	6.46s	44.9s	105s	75.3s
0.05	0.1	21.8s	42.3s	39.3s	77.7s	118s	117s
	0.01	15.4s	38.9s	14.6s	55.8s	95.8s	112s
0.005	0.1	34.1s	47.8s	38.9s	268s	90.5s	172s
	0.01	40.8s	59.5s	45.0s	104s	63.6s	139s

Table 9.2.: Case 2 – Comparison of the FTP, FTPB, and ISSN methods.



(a) Case 1 - Linear problem

(b) Case 1 - Bilinear problem

(c) Case 2 - Linear problem

(d) Case 2 - Bilinear problem

Figure 9.1.: Validation of the theoretical upper bound (Theorem 7.3.14).

Case 3. *(Linear case)* $\Omega = (0,1)^2$, $A = -\Delta$, $u_a \equiv -20$, $u_b \equiv 20$, $z = 1+\sin(2\pi x)\sin(2\pi y) \notin H_0^1(\Omega)$ *and* $f \equiv 1$. *We discretize* Ω *with gridsize* $h = 1/256$. *A is discretized by second-order finite differences.*

Case 4. *(Bilinear case)* $\Omega = (0,1)^2$, $A = -\Delta$, $u_a \equiv -10$, $u_b \equiv 10$, $z = 1 + \sin(2\pi x)\sin(2\pi y) \notin H_0^1(\Omega)$ *and* $f \equiv 1$. *We discretize* Ω *with gridsize* $h = 1/256$. *A is discretized by second-order finite differences.*

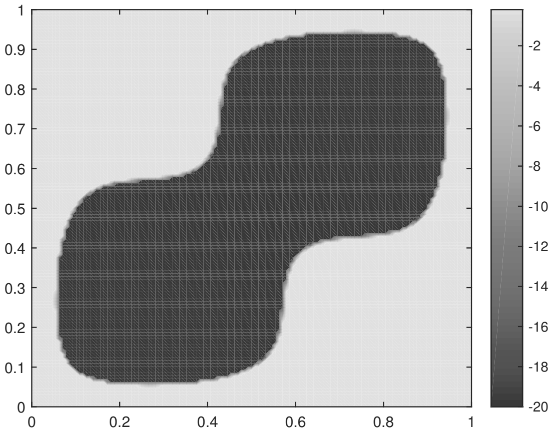
In the Figures 9.2 and 9.3, we present the optimal controls obtained for the Cases 3 and 4, respectively. Notice that the controls obtained with the FTP, FTPB, and ISSN schemes are indistinguishable. We observe that in the case of a small α there is an abrupt change between $u = 0$ and $u = u_b$, whereas for bigger α the change is continuous. We also see that by increasing β the support of u decreases as expected. The different computational times of the FTP, FTPB, and ISSN schemes are given in the figure. We see that the FTPB scheme may outperform the ISSN scheme and vice versa. We also have a case where the ISSN scheme has difficulty to converge; see Figure 9.3, test case (c). Notice that very similar results are also obtained using a globalized version [CB16] of the ISSN scheme. These results and further results of numerical experiments demonstrate that fast truncated proximal schemes represent a valuable alternative to semi-smooth Newton methods.

9.2. Parabolic models

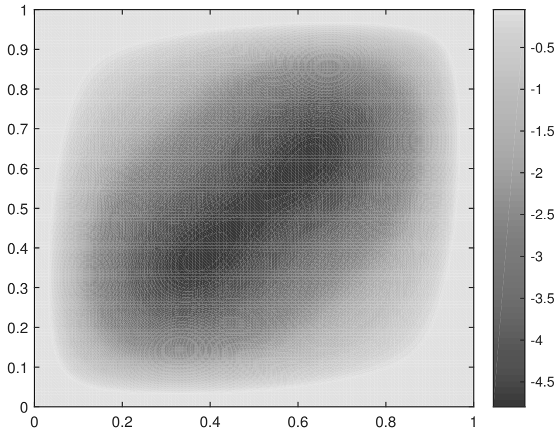
In this section, we present results of numerical experiments to validate the computational performance of our truncated proximal methods applied to parabolic methods and to demonstrate the convergence rate of the proximal residual proved in Theorem 7.3.7. Further, we benchmark our proximal methods with the ISSN scheme discussed in the previous section. For validation purposes, we formulate control problems for which we know the exact solution. We have

Procedure 3. *(Linear control case)*

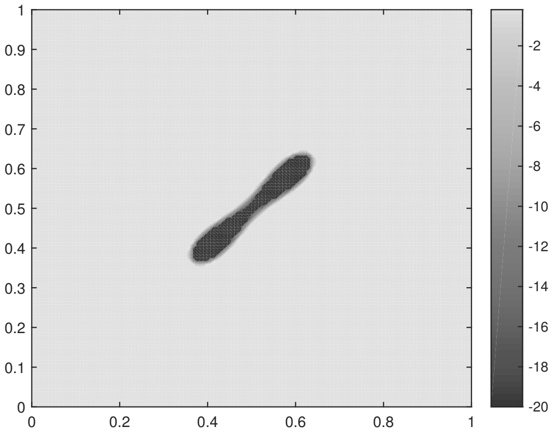
1. *Choose* $\hat{y} \in L^2(0,T;H_0^1(\Omega))$ *and* $\hat{p} \in L^2(0,T;H_0^1(\Omega))$ *arbitrary*
2. *Set* $\hat{u} := \begin{cases} \max\{\frac{-\hat{p}+\beta}{\alpha}, u_a\} & \text{on } \{x \in \Omega_T : \hat{p}(x,t) > \beta\} \\ \min\{\frac{-\hat{p}-\beta}{\alpha}, u_b\} & \text{on } \{x \in \Omega_T : \hat{p}(x,t) < -\beta\} \\ 0 & \text{elsewhere} \end{cases}$.
3. $\mu := -\hat{p} - \alpha\hat{u}$
4. $f := \partial_t y + A\hat{y} + \hat{u}$
5. $z := -\partial_t p + A^*\hat{p} + \hat{y}$



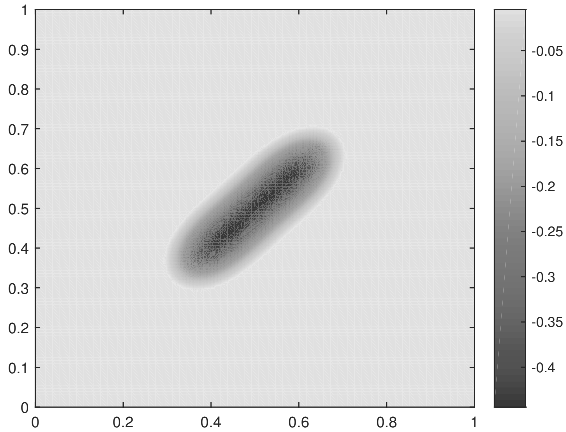
(a) $\beta = 10^{-2},\ \alpha = 10^{-5}$,
FTP: 46.9s FTPB: 82.5s ISSN: 21.5s



(b) $\beta = 10^{-2},\ \alpha = 10^{-2}$
FTP: 2.17s FTPB: 6.07s ISSN: 5.52s

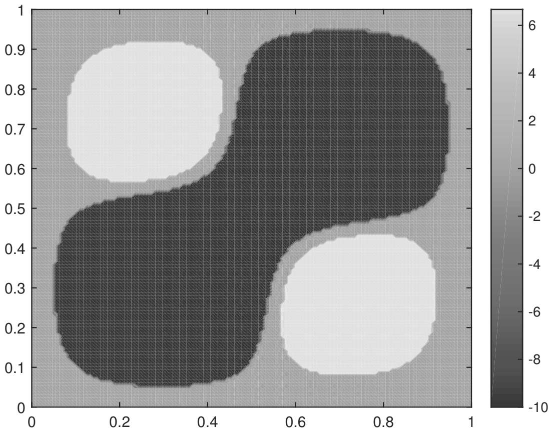


(c) $\beta = 6.5 \cdot 10^{-2},\ \alpha = 10^{-5}$
FTP: 46.5s FTPB: 32.2s ISSN: 17.4s

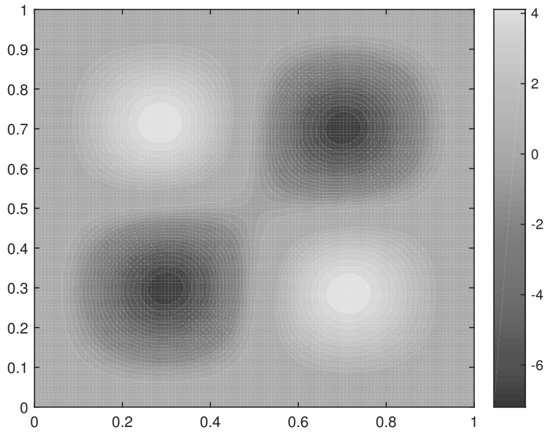


(d) $\beta = 6.5 \cdot 10^{-2},\ \alpha = 10^{-2}$
FTP: 1.81s FTPB: 2.60s ISSN: 3.65s

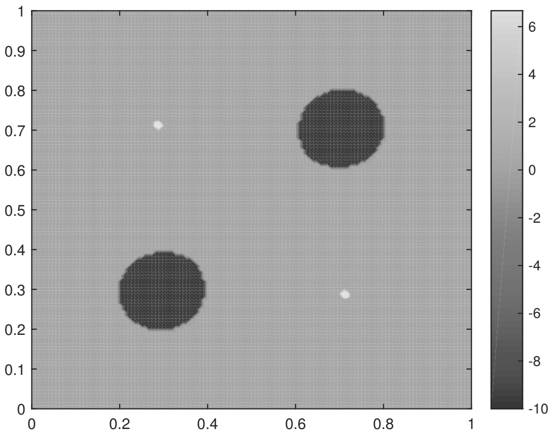
Figure 9.2.: Controls u of Case 3



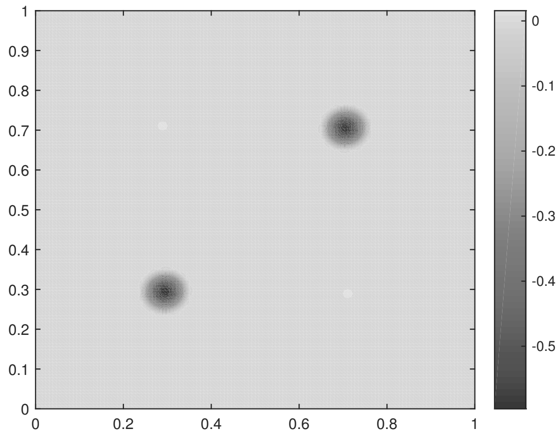
(a) $\beta = 10^{-2},\ \alpha = 10^{-5}$
FTP: 56.7s, FTPB: 47.8s, ISSN: 42.8s



(b) $\beta = 10^{-2},\ \alpha = 10^{-2}$
FTP: 15.3s, FTPB: 8.59s, ISSN: 25.8s



(c) $\beta = 6 \cdot 10^{-2},\ \alpha = 10^{-5}$
FTP: 52.6s, FTPB: 46.1s, ISSN: 1010s



(d) $\beta = 6 \cdot 10^{-2},\ \alpha = 10^{-2}$
FTP: 10.7s, FTPB: 40.9s, ISSN: 6.85s

Figure 9.3.: Controls u of Case 4

Lemma 9.2.1. *Procedure 3 provides a solution $(\hat{y}, \hat{u})$ of the optimal control problem (6.0.2) with the parabolic model and linear control mechanism.*

Proof. We show that the optimality conditions (6.3.32)–(6.3.35) in Theorem 6.3.5 are fulfilled. (6.3.32)–(6.3.34) are obviously fulfilled because of 3.– 5. in routine 3. Now, we consider different cases to show (6.3.35):

- $|\hat{p}| \le \beta$: From 2. we have $\hat{u} = 0$ and from 3. $\mu = -\hat{p}$ and therefore

$$\begin{aligned} B(\hat{u}, \mu) = 0 &- \max\{0, c(-\hat{p} - \beta)\} - \min\{0, c(-\hat{p} + \beta)\} \\ &+ \max\{0, -u_b + c(-\hat{p} - \beta)\} + \min\{0, -u_a + c(-\hat{p} + \beta)\} = 0. \end{aligned}$$

- $\hat{p} > \beta$:
  - ⋆ $u_a \le \frac{-\hat{p}+\beta}{\alpha} \le u_b$: From 2. we have $\hat{u} = \frac{-\hat{p}+\beta}{\alpha} < 0$ and from 3. we have $\mu = -\hat{p} - \alpha\hat{u} = -\beta$, therefore

$$\begin{aligned} B(\hat{u}, \mu) &= \hat{u} - \max\{0, \hat{u} - c(2\beta)\} - \min\{0, \hat{u})\} \\ &\quad + \max\{0, \hat{u} - u_b - c(2\beta)\} + \min\{0, \hat{u} - u_a\} \\ &= \hat{u} - 0 - \hat{u} + 0 + 0 = 0. \end{aligned}$$

  - ⋆ $\frac{-\hat{p}+\beta}{\alpha} \le u_a$: From 2. we have $\hat{u} = u_a$ and from 3. we have $\mu = -\hat{p} - \alpha u_a$, therefore

$$\begin{aligned} B(\hat{u}, \mu) &= u_a - \max\{0, u_a + c(-\hat{p} - \alpha u_a - \beta)\} - \min\{0, u_a + c(-\hat{p} - \alpha u_a + \beta)\} \\ &\quad + \max\{0, u_a - u_b + c(-\hat{p} - \alpha u_a - \beta)\} \\ &\quad + \min\{0, u_a - u_a + c(-\hat{p} - \alpha u_a + \beta)\} \\ &= u_a - 0 - (u_a + c(-\hat{p} - \alpha u_a + \beta)) + 0 + c(-\hat{p} - \alpha u_b + \beta) = 0. \end{aligned}$$

- $\hat{p} < -\beta$
  - ⋆ $u_a \le \frac{-\hat{p}-\beta}{\alpha} \le u_b$: From 2. we have $\hat{u} = \frac{-\hat{p}-\beta}{\alpha} > 0$ and from 3. we have $\mu = -\hat{p} - \alpha\hat{u} = \beta$. So

$$\begin{aligned} B(\hat{u}, \mu) &= \hat{u} - \max\{0, \hat{u}\} - \min\{0, \hat{u} + c(2\beta)\} \\ &\quad + \max\{0, \hat{u} - u_b\} + \min\{0, \hat{u} - u_a + c(2\beta)\} \\ &= \hat{u} - \hat{u} - 0 + 0 + 0 = 0. \end{aligned}$$

  - ⋆ $\frac{-\hat{p}-\beta}{\alpha} \ge u_b$: From 2. we have $\hat{u} = u_b$ and from 3. we have $\mu = -\hat{p} - \alpha u_b$, therefore

$$\begin{aligned} B(\hat{u}, \mu) &= u_b - \max\{0, u_b + c(-\hat{p} - \alpha u_b - \beta)\} - \min\{0, u_b + c(-\hat{p} - \alpha u_b + \beta)\} \\ &\quad + \max\{0, u_b - u_b + c(-\hat{p} - \alpha u_b - \beta)\} \\ &\quad + \min\{0, u_b - u_a + c(-\hat{p} - \alpha u_b + \beta)\} \\ &= u_b - (u_b + c(-\hat{p} - \alpha u_b - \beta)) - 0 + c(-\hat{p} - \alpha u_b - \beta) + 0 = 0. \end{aligned}$$

□

Procedure 4. *(Bilinear control case)*

1. *Choose* $\hat{y} \in L^2(0,T;H_0^1(\Omega))$ *and* $\hat{p} \in L^2(0,T;H_0^1(\Omega))$ *arbitrary*
2. *Set* $\hat{u} := \begin{cases} \max\{\frac{-\hat{p}\hat{y}+\beta}{\alpha}, u_a\} & \text{on } \{x \in \Omega_T : \hat{p}(x,t)\hat{y}(x,t) > \beta\} \\ \min\{\frac{-\hat{p}\hat{y}-\beta}{\alpha}, u_b\} & \text{on } \{x \in \Omega_T : \hat{p}(x,t)\hat{y}(x,t) < -\beta\} \\ 0 & \text{elsewhere} \end{cases}.$
3. $\mu := -\hat{p}\hat{y} - \alpha\hat{u}$
4. $f := \partial_t y + A\hat{y} + \hat{u}\hat{y}$
5. $z := -\partial_t p + A^*\hat{p} + \hat{y} + \hat{u}\hat{p}$

Lemma 9.2.2. *Procedure 4 provides a solution* $(\hat{y}, \hat{u})$ *to the optimal control problem (6.0.2) with the parabolic model and bilinear control mechanism.*

Proof. The proof is similar to the one of the linear case. □

Next, we specify the parabolic operator, the domain of computation, the choice of $\hat{y}$ and $\hat{p}$, and some optimization and numerical parameters. We consider the following test case.

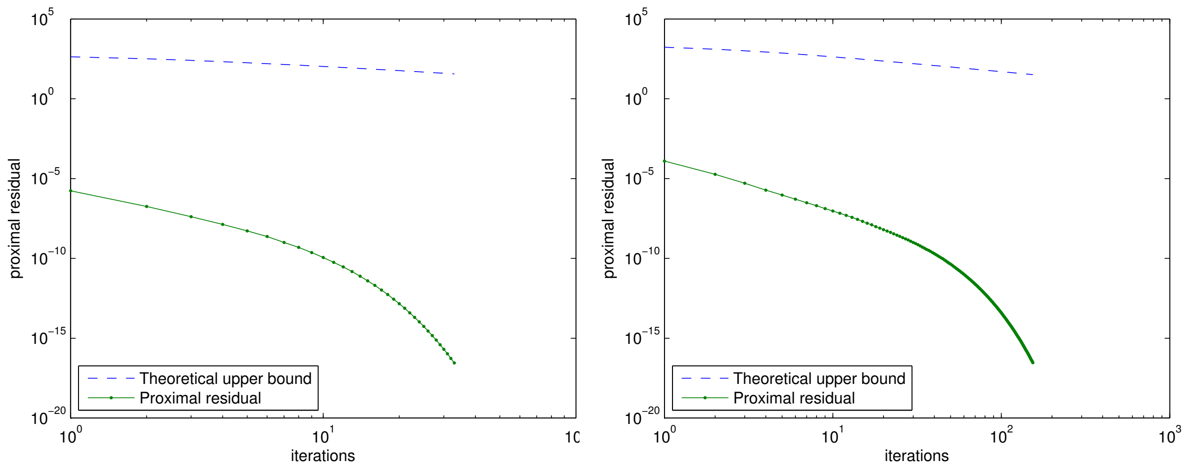
Case 5. $\Omega = (0,1)^2$, $T = 1$, $A = -\Delta$, $\hat{y} = 5\sqrt{\beta}t\sin(3\pi x_1)\sin(\pi x_2)$, $\hat{p} = 5\sqrt{\beta}(t-1)\sin(\pi x_1)\sin(\pi x_2)$, $u_a \equiv -1$ *and* $u_b = 2$. *The functions* f *and* z *are then given by Procedure 3, resp. Procedure 4. We discretize* Ω *with gridsize* $h = 1/32$ *and* $\delta t = 1/1024$. *A is discretized by second-order finite differences and the time derivative is discretized by finite forward differences. The results are shown in Table 9.3.*

The high temporal resolution is used to reduce the error of calculating the functional in the VTIP method. However, in each step of the CTIP and the ISSN method, the functional is not needed and the algorithms also converge for smaller temporal resolution. We compare the CTIP, VTIP, and ISSN schemes in terms of computational time. In the CIIP method, we calculate the smallest Lipschitz constant as the dominant eigenvalue of $\nabla^2 \hat{J}_2(u)$ with a power iteration. The effort of this calculation is included in the total CPU time. This power iteration is stopped if the difference between two iterates of the norm $\|\nabla^2 \hat{J}_2\|_{L^2,L^2}$ is less or equal than a tolerance of 10^{-5}. For the VTIP method, we use backtracking with $\eta = 1.5$ and $L_0 = 0.0005$. All algorithms are stopped if $\|B(u_k, \mu_k)\| < 10^{-6}$. Furthermore, we used $c_2 = 10^{-3}$, $\theta = 0.5$, $\varepsilon_k = \frac{1}{(k+1)^3}$ and the stepsize s was chosen by $s = 1.9\frac{1-\theta}{L+2c_2}$. We can see in Table 9.3 that the CTIP and VTIP methods result competitive to the ISSN method. In the case of a big α, the proximal methods outperform the ISSN scheme, while in the case of a sufficiently small α, the ISSN performs better.

In order to validate the theoretical rate of convergence of the proximal residual, the theoretical upper bound of Theorem 7.3.7 and the actual error of the proximal residual in correspondence to Case 5, with $\beta = 0.1$ and $\alpha = 0.001$, are plotted in Figure 9.1. We see that the actual convergence may even be faster than the theoretical prediction.

		linear case			bilinear case		
α	β	CTIP	VTIP	ISSN	CTIP	VTIP	ISSN
0.01	0.1	98.3s	126s	102s	219s	288s	501s
	0.01	99.2s	130s	131s	163s	227s	330s
0.001	0.1	69.4s	96.3s	114s	517s	225s	794s
	0.01	78.8s	107s	128s	172s	192s	514s
0.0001	0.1	338s	530s	97.1s	8327s	1331s	1077s
	0.01	368s	444s	141s	710s	521s	812s

Table 9.3.: Case 5 – Comparison of the CTIP, VTIP and ISSN methods.



(a) Case 5 - Linear problem.

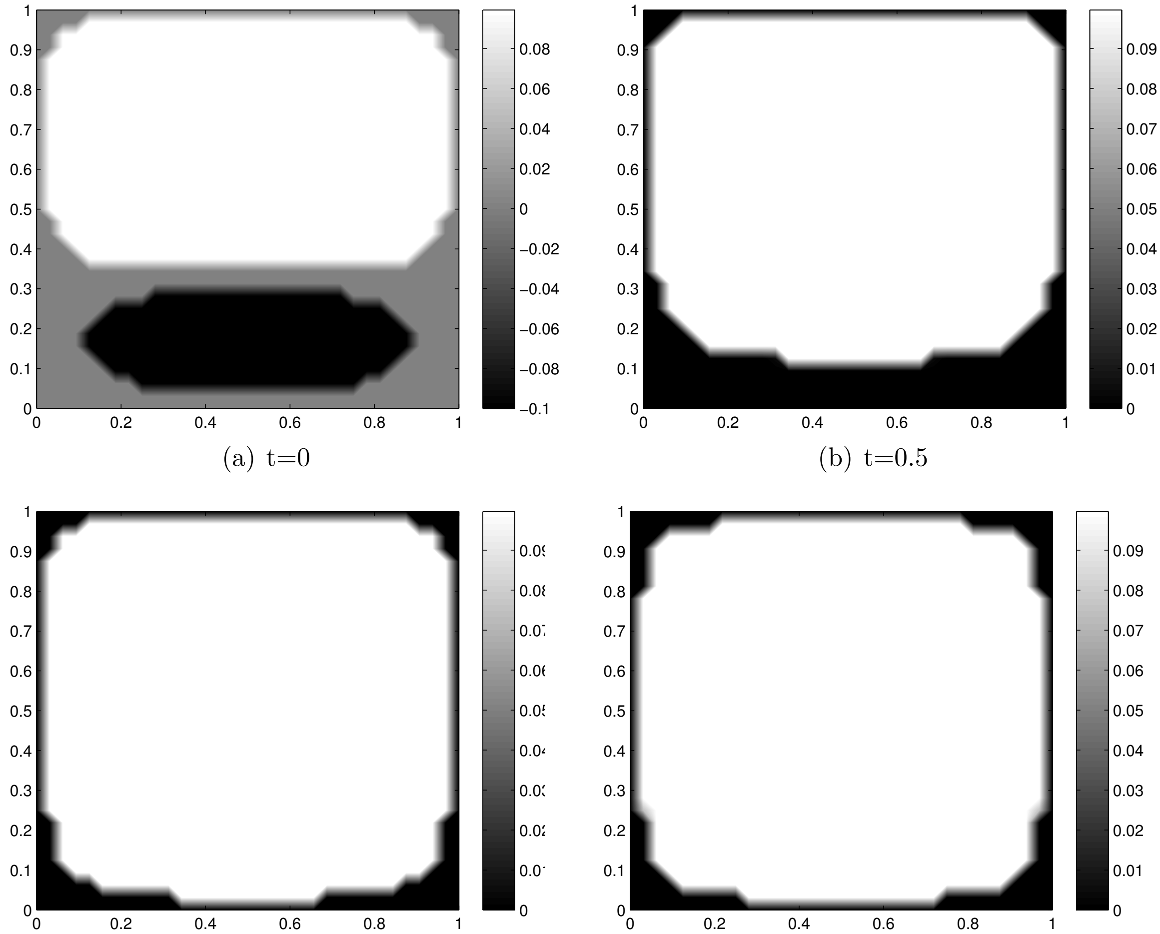
(b) Case 5 - Bilinear problem.

Figure 9.4.: Validation of the theoretical upper bound (Theorem 7.3.7).

We conclude this section considering challenging parabolic linear bilinear control cases where the exact solution is not known. In these cases, the target function is not attainable. We have

Case 6. $\Omega = (0,1)^2$, $T = 1$, $A = -\Delta$, $u_a \equiv -0.1$, $u_b \equiv 0.1$, $z = (1-t)\sin(\pi x_1)\sin(2\pi x_2)$, $f \equiv 5$ *and* $y_0 = \sin(\pi x_1)\sin(2\pi x_2)$. *We discretize* Ω *with gridsize* $h = 1/32$ *and* $\delta t = 1/1024$. *A is discretized by second-order finite differences and the time derivative is discretized by finite forward differences.*

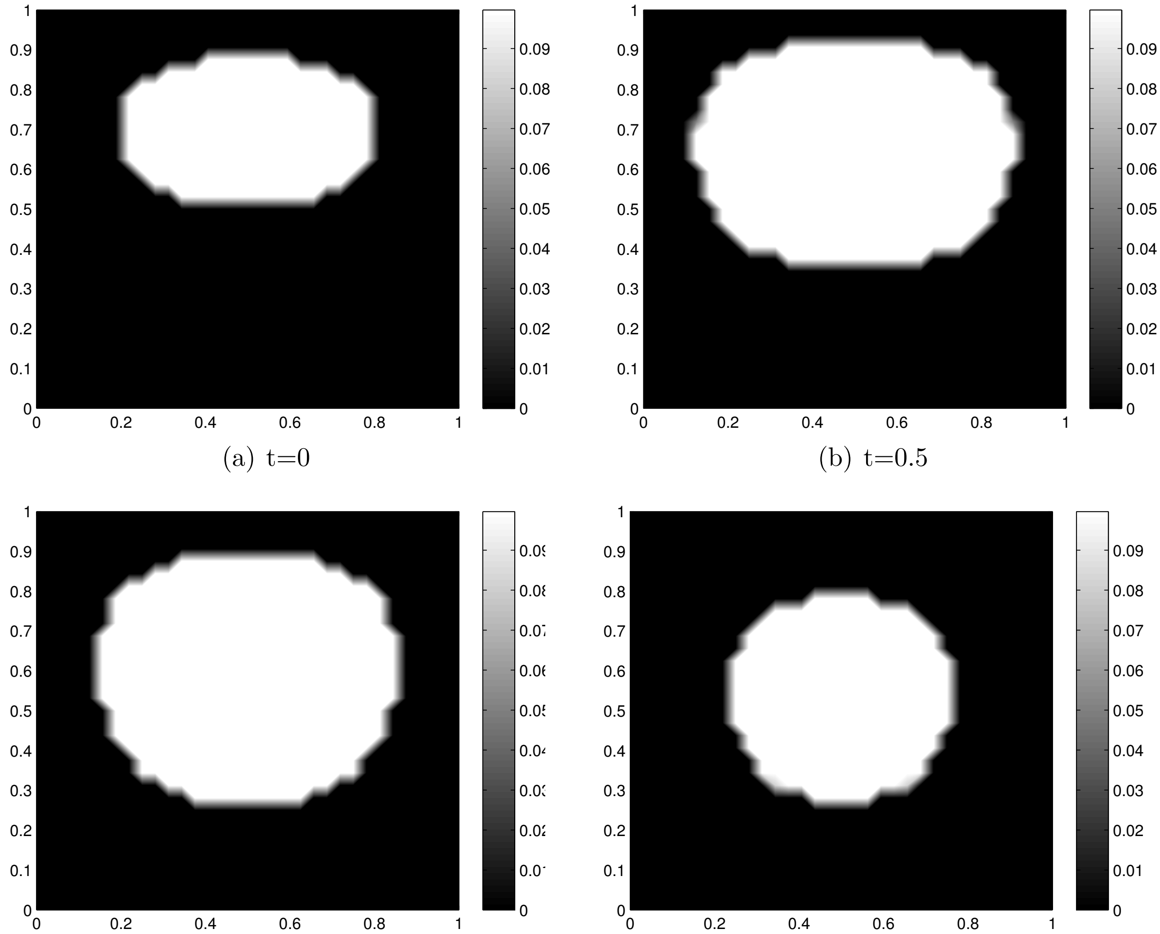
In the Figures 9.5 - 9.10, we depict the optimal controls obtained for Case 6 in the linear and bilinear cases, respectively. Notice that the controls obtained with the CTIP, VTIP, and ISSN schemes are indistinguishable. We can see that choosing smaller values of α, sharper edges between the regions $u = 0$ and $u = u_a$ and $u = u_b$ appear. We also see that by increasing β the support of u decreases as expected. The different computational times of the CTIP, VTIP, and ISSN schemes are also shown in the figures. We obtain the same dependence as for Case 3 & 4 of the computational performance with respect the optimization parameters. These results and further results of numerical experiments demonstrate that fast truncated proximal schemes represent a valuable alternative to the state-of-the-art semi-smooth Newton schemes.



(a) t=0 (b) t=0.5

(c) t=0.75 (d) t=0.875

Figure 9.5.: Controls u of Case 6 with linear control mechanism, $\beta = 10^{-3}$, $\alpha = 10^{-4}$. CTIP: 354s, VTIP: 200s, ISSN: 259s.



(a) t=0 (b) t=0.5

(c) t=0.75 (d) t=0.875

Figure 9.6.: Controls u of Case 6 with linear control mechanism, $\beta = 10^{-2}$, $\alpha = 10^{-4}$. CTIP: 300s, VTIP: 178s, ISSN: 254s.

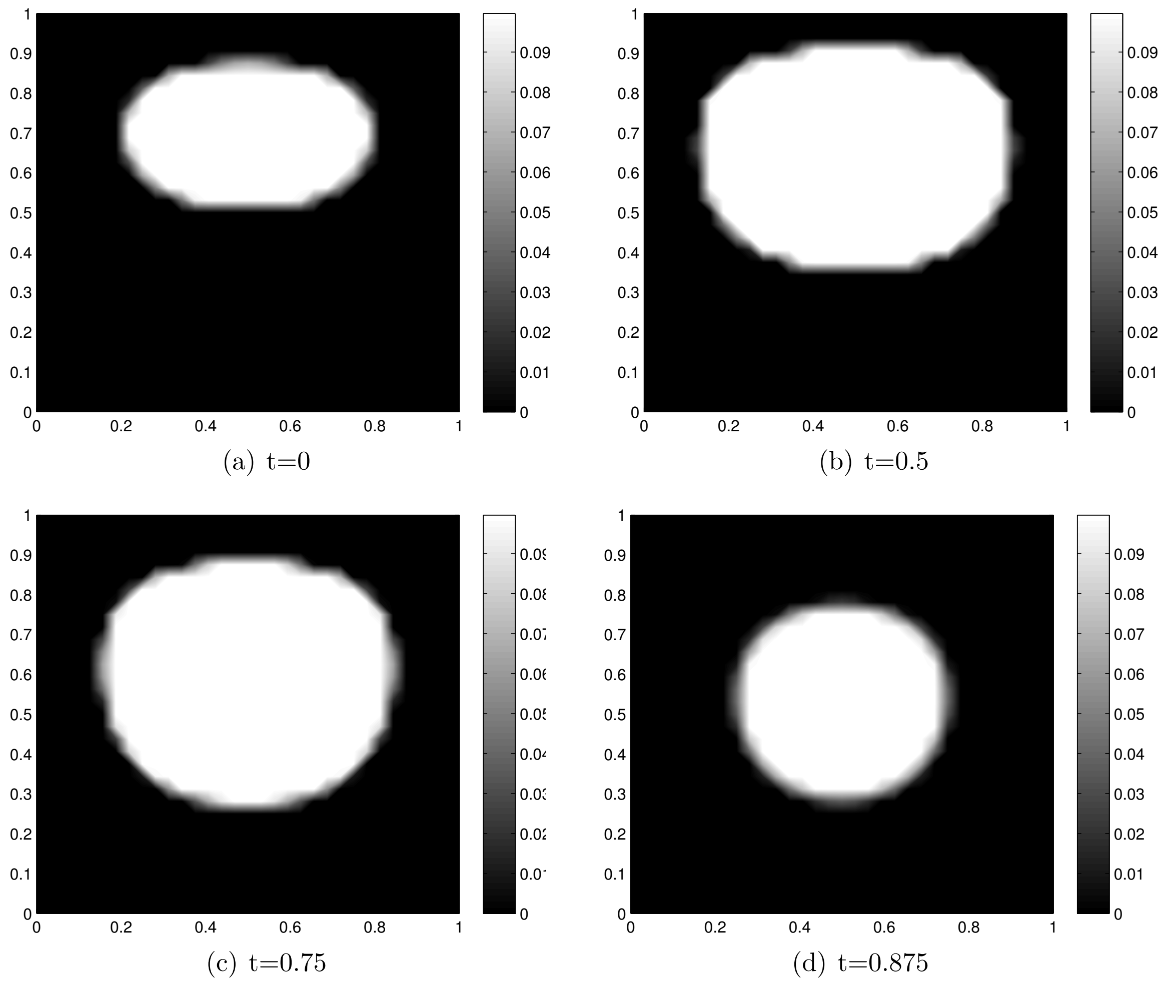
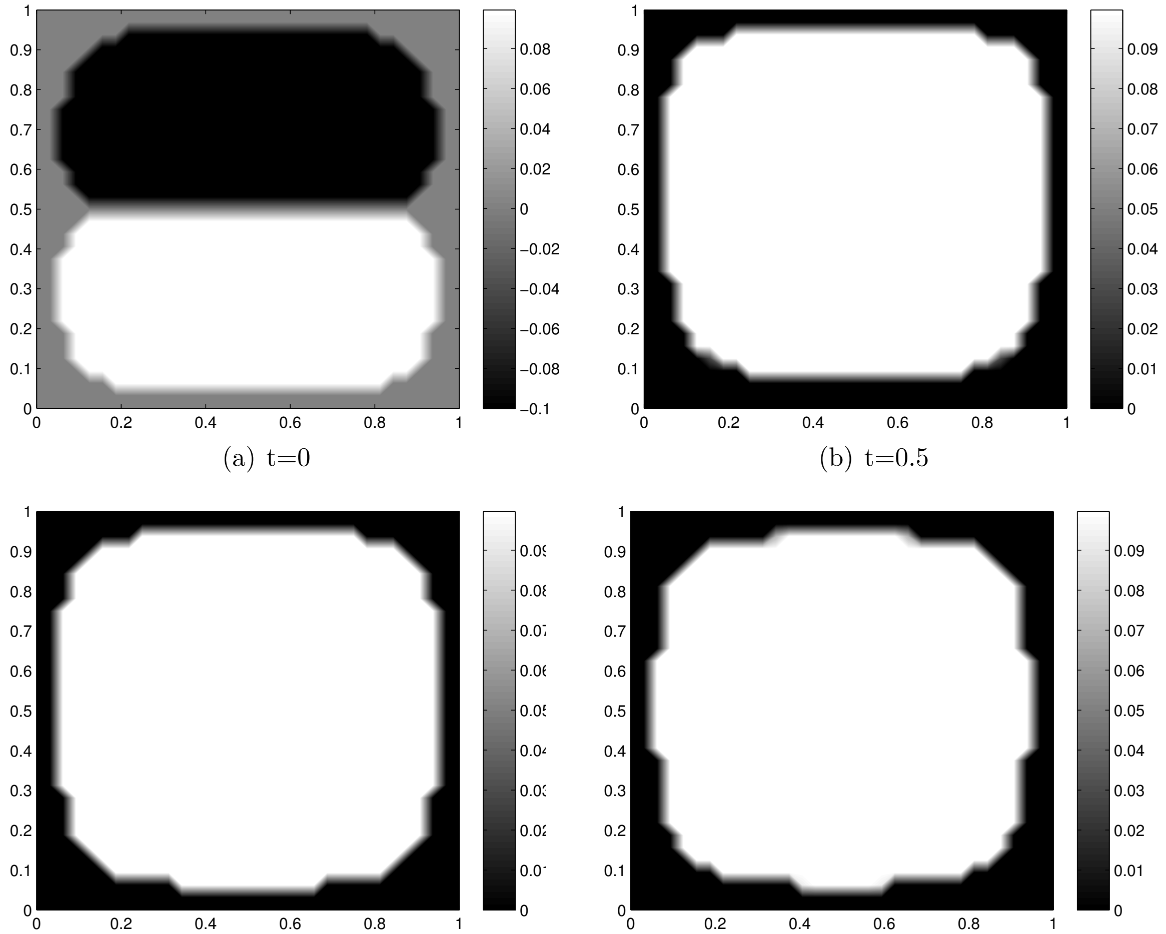


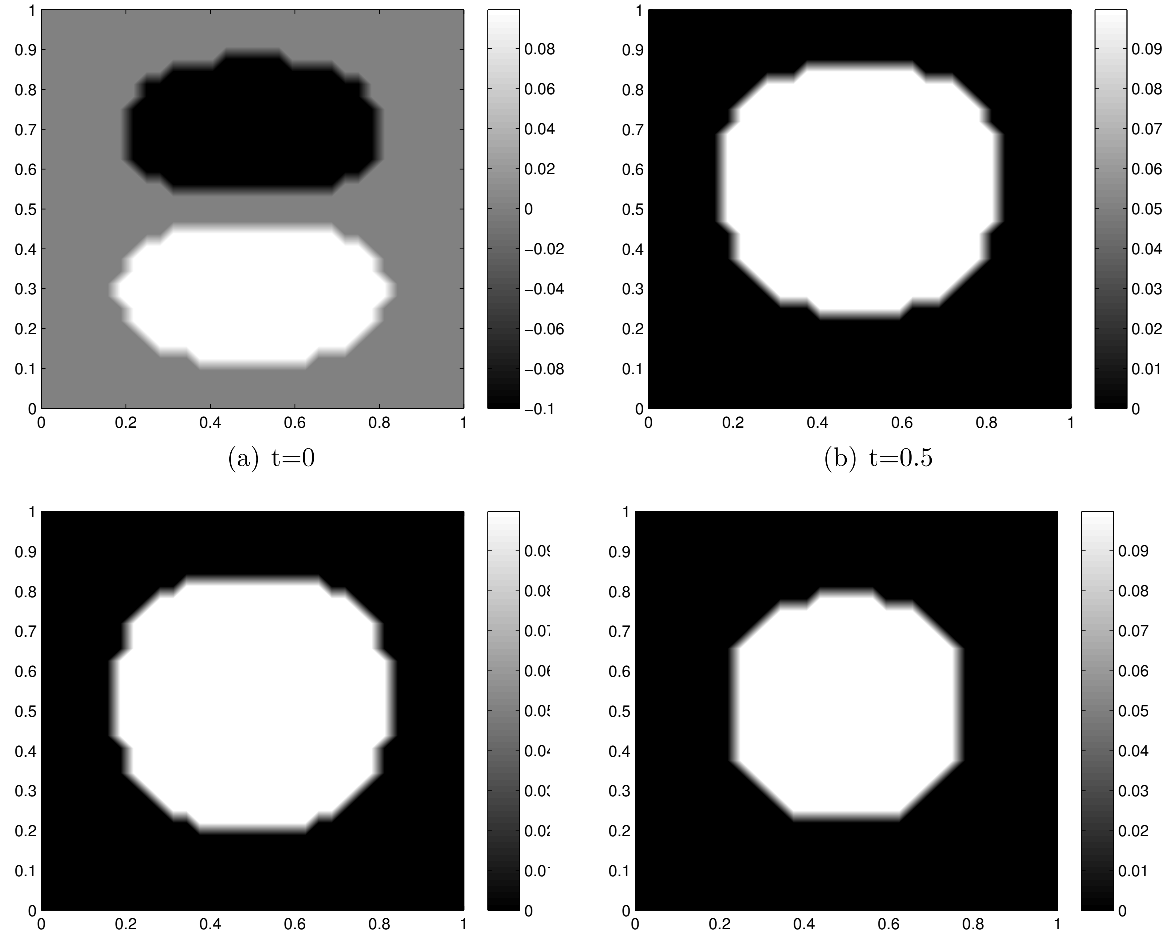
Figure 9.7.: Controls u of Case 6 with linear control mechanism, $\beta = 10^{-2}$, $\alpha = 10^{-2}$. CTIP: 117s, VTIP: 138s, ISSN: 82s.



(a) t=0 (b) t=0.5

(c) t=0.75 (d) t=0.875

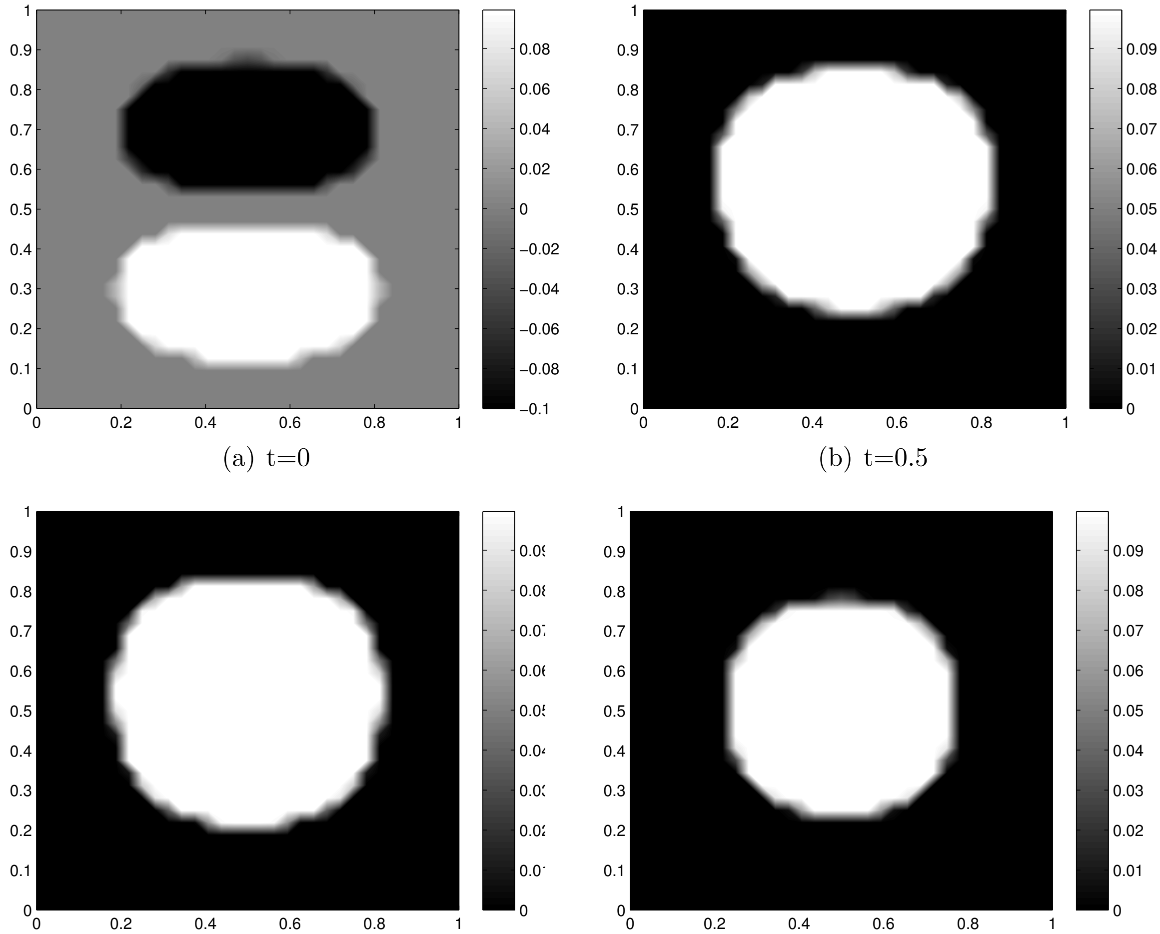
Figure 9.8.: Controls u of Case 6 with bilinear control mechanism, $\beta = 10^{-3},\ \alpha = 10^{-4}$.
CTIP: 566s, VTIP: 262s, ISSN: 249s.



(a) t=0 (b) t=0.5

(c) t=0.75 (d) t=0.875

Figure 9.9.: Controls u of Case 6 with bilinear control mechanism, $\beta = 10^{-2},\ \alpha = 10^{-4}$.
CTIP: 350s, VTIP: 187s, ISSN: 330s.



(a) t=0

(b) t=0.5

(c) t=0.75

(d) t=0.875

Figure 9.10.: Controls u of Case 6 with bilinear control mechanism, $\beta = 10^{-2},\ \alpha = 10^{-2}$. CTIP: 118s, VTIP: 163s, ISSN: 229s.

10. Conclusion

First-order proximal schemes were discussed for finite dimensional and infinite dimensional applications. In finite dimensions they were used for solving l_1- and TV-minimization problems in image reconstruction with successful application to MRI. Convergence of these methods in this setting was proved.

In infinite dimensions, first-order proximal schemes were used for solving nonsmooth linear and bilinear elliptic and parabolic optimal control problems. A complete analysis of these methods was presented and convergence of the function values as well as the existence of a sequence that converges to a solution was proven. Furthermore, it was shown that the proximal residual has convergence rate of $\mathcal{O}(1/\sqrt{k})$, resp. $\mathcal{O}(1/k^2)$, in the fast case. For benchmarking purposes, the proposed truncated proximal schemes were compared to an inexact semismooth Newton method. Results of numerical experiments demonstrated the computational effectiveness of truncated proximal schemes and successfully validated the theoretical estimates.

List of Figures

List of Tables

List of Algorithms

A. Matlab Code

This PhD thesis is completed with a CD-ROM containing MATLAB codes for solving the elliptic and parabolic control problems. To run the elliptic control solver type in the MATLAB environment

» `[u1,u2,u3]=CodeElliptic;`

For the parabolic case type

» `[u1,u2,u3]=CodeParabolic;`

The parabolic solver is shown in Listing A.1.

Listing A.1: A parabolic optimal control example

```
%% Main function;
function [u1,u2,u3] = CodeParabolic
%        RETURN:    u1     [N,N,NTime] Optimal Control of CTIP method
%                   u2     [N,N,NTime] Optimal Control of VTIP method
%                   u3     [N,N,NTime] Optimal Control of ISSN method
%        This script is solving the parabolic optimal control problem
%        min 1/2||y-z||^2+alph/2||u||^2+bet||u||_L1
%        subject to
%        y_t-laplace y+u =f        in [0,T] x Omega  (lin=0)
% resp.  y_t-laplace y+uy=f        in [0,T] x Omega  (lin=1)
%        y=y0                      on t=0 x Omega
%        y=0                       on [0,T] x delltaOmega
%        low<=u<=up

%   First-optimize-then-discretize strategy  %

close all

% Define the global variables:

% dimensions:
global N
global NTime
% parameters:
global bet
global alph
% bounds:
```

```
global low
global up
% lin=0 (linear control), lin=1(bilinear control)
global lin
%tolerance
global tol_stop

%% Set the global variables:

% dimensions:
Nexp            = 5;
N               = 2^Nexp+1;
NTime           = 2^(Nexp+5)+1;
% parameters:
bet             = 1e-1;
alph            = 1e-2;
% bounds:
low             = -1;
up              = 2;
% lin=0 (linear control), lin=1 (bilinear control)
lin=1;
% tolerance:
tol_stop        = 1e-6;

% Getting the test constants z,f and y0
[z,f,y0,u_test] = GetSystem;

% Initializing the optimization variable:
u0              = zeros(N,N,NTime);
u=u0;

% maximum iterations:
maxit           = 1000;

% Exact functional for comparison
f_ex            = functional( u_test,f, z,y0);

%% CTIP
tic
L = powerit(50,f,z,y0,u0);
[u1] = CTIP(f,z,y0,u0,L,maxit);
toc

%% VTIP
tic
[u2] = VTIP(f,z,y0,u0,maxit);
```

```
toc

%% Semismooth Newton
tic
[u3] = ssnewton(f,z,y0,u0,maxit);
toc

end


%% test problem
function [z,f,y0,u_test] = GetSystem
%       RETURN:         z       [N,N,NTime]     target state
%                       f       [N,N,NTime]     right hand side
%                       y0      [N,N]           starting state
%                       u_test  [N,N,NTime]     test control

%       This function creates a test problem including the solution


% Global variables:
global h
global bet
global alph
global low
global up
global lin
global N
global NTime

% Define the gridpoints:
T                       = 1;
h                       = T / (N-1);
h2=h*h;
time_max = 1;
dtime       = time_max / (NTime-1);
[x_vec,y_vec, time_vec]=meshgrid( 0:h:1, 0:h:1,0:dtime:time_max);

% Discretize the Laplace operator:
A = gallery('poisson',N)/h2;


if lin==0 % linear control mechanism
    % example:
    y_test=5*sqrt(bet)*time_vec.*sin(3*pi*x_vec).*sin(pi*y_vec);
    p_test=5*sqrt(bet)*(time_vec-1).*sin(pi*x_vec).*sin(pi*y_vec);
```

```

    % Calculating the optimization variable
    u_test=zeros(N,N,NTime);
    ind = p_test>bet;
    u_test(ind)=(-p_test(ind)+bet)/alph;
    ind=p_test<-bet;
    u_test(ind)=(-p_test(ind)-bet)/alph;
    u_test=max(low,min(u_test,up));

    % Calculating the derivative of the states with respect to time
    deltay=y_test;
    deltay(:,:,2:end)=(y_test(:,:,2:end)-y_test(:,:,1:end-1))/dtime;
    deltay(:,:,1)=y_test(:,:,2);
    deltay=reshape(deltay,[N*N,NTime]);
    deltap=p_test;
    deltap(:,:,1:end-1)=(p_test(:,:,2:end) -...
        p_test(:,:,1:end-1))/dtime;
    deltap(:,:,end)=-p_test(:,:,end-1);
    deltap=reshape(deltap,[N*N,NTime]);

    % Calculating the right-hand-side and the tartget-state
    f=deltay+A*reshape(y_test,[N*N,NTime])+...
        reshape(u_test,[N*N,NTime]);
    z=-deltap+A*reshape(p_test,[N*N,NTime])+...
        reshape(y_test,[N*N,NTime]);
    f=reshape(f,[N,N,NTime]);
    z=reshape(z,[N,N,NTime]);

else % bilinear control mechanism
    % example:
    y_test=5*sqrt(bet)*time_vec.*sin(3*pi*x_vec).*sin(pi*y_vec);
    p_test=5*sqrt(bet)*(time_vec-1).*sin(pi*x_vec).*sin(pi*y_vec);

    % Calculating the optimization variable
    u_test=zeros(N,N,NTime);
    ind=(p_test.*y_test)>bet;
    u_test(ind)=(-(y_test(ind).*p_test(ind))+bet)/alph;
    ind=(y_test.*p_test)<-bet;
    u_test(ind)=(-(y_test(ind).*p_test(ind))-bet)/alph;
    u_test=max(low,min(u_test,up));

    %Calculating the derivative of the states with respect to time
    deltay=y_test;
    deltay(:,:,2:end)=(y_test(:,:,2:end)-y_test(:,:,1:end-1))/dtime;
    deltay(:,:,1)=y_test(:,:,2);
    deltay=reshape(deltay,[N*N,NTime]);
```

```
    deltap=p_test;
    deltap(:,:,1:end-1)=(p_test(:,:,2:end)-...
        p_test(:,:,1:end-1))/dtime;
    deltap(:,:,end)=-p_test(:,:,end-1);
    deltap=reshape(deltap,[N*N,NTime]);

    % Calculating the right-hand-side and the tartget-state
    f=deltay+A*reshape(y_test,[N*N,NTime])+...
        reshape(u_test,[N*N,NTime]).*reshape(y_test,[N*N,NTime]);
    z=-deltap+A*reshape(p_test,[N*N,NTime])+...
        reshape(y_test,[N*N,NTime])+...
        reshape(u_test.*p_test,[N*N,NTime]);
    f=reshape(f,[N,N,NTime]);
    z=reshape(z,[N,N,NTime]);
end
% Calculating the starting state
y0=y_test(:,:,1);
end

%% cost functional
function [ func ] = functional( u, f, z,y0 )
%       INPUT:          u       [N,N,NTime]     control
%                       f       [N,N,NTime]     right-hand-side
%                       z       [N,N,NTime]     target state
%                       y0      [N,N]           starting state
%       RETURN:         func    [1]             functional value

% This function calculates the actual functional value

global alph
global bet
global lin
global N
global NTime

% Calculating the state:
if lin==0 % linear control mechanism
    [y,~]=laplacesolv(-u+f,u,z,y0,1e-8,u,u,0);
else % bilinear control mechanism
    [y,~]=laplacesolv(f,u,z,y0,1e-8,u,u,0);
end

% Getting function handles of the L1-norm and the L2-norm
% using the trapez rule
x=0:1/(N-1):1;
x2=0:1/(NTime-1):1;
```

```
normL2sq=@(uu)trapz(x2,trapz(x,trapz(x,uu.^2)));
normL1=@(uu)trapz(x2,trapz(x,trapz(x,abs(uu))));

% Calculating the functional value
func =  0.5*normL2sq(y-z)+alph*0.5  *normL2sq(u)+bet * normL1(u);
end

%% gradient assembler;
function [ grad ] = gradient( u,y0,p0 )
%       INPUT:          u       [N,N,NTime]     control
%                       y0      [N,N,NTime]     state
%                       p0      [N,N,NTime]     adjoint state
%       RETURN:         grad    [N,N,NTime]     gradient

% This function calculates the actual gradient

% Getting the necessary global variables
global alph
global lin

% Calculating the gradient
if lin==0 % linear control mechanism
    grad=alph*u+p0;
else % bilinear control mechanism
    grad=alph*u+y0.*p0;
end

end

%%% Calculating Lipschitz constant
function Lu = powerit(it,f,z,y0,u0)
%       INPUT:          it      [1]             maximum iterations
%                       f       [N,N,NTime] right-hand side
%                       z       [N,N,NTime] target state
%                       y0      [N,N]           starting state
%                       u0      [N,N,NTime] control
%       RETURN:         Lu      [1]             Lipschitz constant

% This function calculates the Lipschitz constant
% using a power iteration

% Getting the necessary global variables
global alph
global lin

%Setting the tolerances
```

```
tol=1e-6;
tol2=1e-4;

% Initializing the start values of the power iteration
u = ones(size(f));
Lu=u;
y=u;
p=u;

% Starting power iteration
for i = 1:it
    Luold=Lu;

    if lin==0 % linear control mechanism
        % Calculating the Hessian
        temp=laplacesolv2(u,u0,zeros(size(y0)),tol,y,0);
        hess = alph*u+laplacesolv2(temp,u0,zeros(size(y0)),tol,y,0);
    else  % bilinear control mechanism
        % Calculating the Hessian
        [y,p]=laplacesolv(f,u,z,y0,tol,y,p,1);
        yprime=laplacesolv2(-y.*u,u,zeros(size(y0)),tol,y,0);
        pprime=laplacesolv2(-p.*u-yprime,u,zeros(size(y0)),tol,p,1);
        hess=alph*u+(y.*pprime+yprime.*p);
    end

    % Updating the Hessian
    Lu = norm(hess(:));
    u = hess/Lu;

    % Stopping if step is small enough:
    if norm(Lu(:)-Luold(:))<tol2
        break
    end
end
end

function [u] = ssnewton(f,z,y0,u0,maxit)
%       INPUT:      f       [N,N,NTime] right-hand side
%                   z       [N,N,NTime] target state
%                   y0      [N,N]       starting state
%                   u0      [N,N,NTime] control
%                   maxit   [1]         maximum number of iterations
%       RETURN:     u       [N,N,NTime] optimal control

% This function implements the inexact semismooth Newton method

```

```
% Initializing the global variables
global alph
global bet
global low
global up
global h
global tol_stop
global lin
global N
global NTime

% Getting dimension
n2=N*N*NTime;

% starting point
u = u0;
x0=zeros(size(u));
y=x0;
p=x0;

% tolerance for the truncation
tol_ex=1e-6;

% Initialize iterations and stopping criterion
it=1;
stopping=1;
delta=x0;

% Getting the states
if lin==0
    [y,p]=laplacesolv(-u+f,u,z,y0,tol_ex,y,p,1);
else
    [y,p]=laplacesolv(f,u,z,y0,tol_ex,y,p,1);
end

% Starting iteration
while (stopping>tol_stop)&&(it<=maxit)
    if lin==0 %linear control mechanism
        % Getting functional F(u)=0
        F=u+1./alph*(-max(0,-p-bet)-min(0,-p+bet)...
            +max(0,-p-bet-alph*up)+min(0,-p+bet-alph*low));

        % Getting the generalized derivative
        ind=((-p+bet<=0 & -p+bet>alph*low) |...
                (-p-bet<alph*up & -p-bet>=0));
        ind=ind(:);
```

```
        temp=@(x)laplacesolv2(x,u,zeros(size(y0)),tol_ex,y,0);
        G=@(x)x+1./alph*(sparse(1:n2,1:n2,ind,n2,n2,n2)*...
                reshape(temp(temp(reshape(x,[N,N,NTime]))),[n2,1]));

        % Getting the tolerance depending on the functional value
        tol_ex2=norm(F(:))/(it+1);
        if tol_ex2>1
            tol_ex2=0.5;
        end

        % Calculating the step
        [T,delta]=evalc('gmres(G,F(:),[],tol_ex2,N,[],[],delta(:))');
        delta=reshape(delta,[N,N,NTime]);

        % Updating the optimization variable
        u=u-delta;
    else % bilinear control mechanism
        % Calculating the functional F(u)=0
        py=p.*y;
        F=u+1./alph*(-max(0,-py-bet)-min(0,-py+bet)...
            +max(0,-py-bet-alph*up)+min(0,-py+bet-alph*low));

        % Getting the generalized derivative
        ind=((-py+bet<=0 & -py+bet>alph*low) |...
                (-py-bet<alph*up & -py-bet>=0));
        ind=ind(:);
        if lin==0
            [y,p]=laplacesolv(-u+f,u,z,y0,tol_ex,y,p,1);
        else
            [y,p]=laplacesolv(f,u,z,y0,tol_ex,y,p,1);
        end
        yprime=@(x)laplacesolv2(-y.*x,u,zeros(size(y0)),tol_ex,y,0);
        pprime=@(x)laplacesolv2(-p.*x-yprime(x),u,...
                        zeros(size(y0)),tol_ex,p,1);
        hess=@(x)y.*pprime(x)+yprime(x).*p;
        G=@(x)x+1./alph*(sparse(1:n2,1:n2,ind,n2,n2,n2)*...
                reshape(hess(reshape(x,[N,N,NTime])),[n2,1]));

        % Getting the tolerance depending on the functional value
        tol_ex2=norm(F(:))/(it+1)^2;
        if tol_ex2>1
            tol_ex2=0.5;
        end

        % Calculating the step
        [T,delta]=evalc('gmres(G,F(:),[],tol_ex2,N,[],[],delta(:))');
```

```
            delta=reshape(delta,[N,N,NTime]);

            % Updating the optimization variable
            u=u-delta;
        end

        % Calculating the stopping criterion
        if lin==0
            [y,p]=laplacesolv(-u+f,u,z,y0,tol_ex,y,p,1);
            mu=-p-alph*u;
        else
            [y,p]=laplacesolv(f,u,z,y0,tol_ex,y,p,1);
            mu=-y.*p-alph*u;

        end
        B=u-max(0,u+mu-bet)-min(0,u+mu+bet)+max(0,u-(up)+mu-bet)+...
            min(0,u-(low)+mu+bet);
         stopping=1/sqrt(h*(length(f)-1))*norm(B(:));

        % Calculating the funcional value
        func = functional( u, f, z,y0 );

        % Plot
        fprintf('it = %d stop = %.3e func = %.10e \n', ...
            it, stopping , func );

        it=it+1;
end
end

%% VTIP method
function [u] = VTIP(f,z,y0,u0,maxit)
%        INPUT:     f       [N,N,NTime] right-hand side
%                   z       [N,N,NTime] target state
%                   y0      [N,N]       starting state
%                   u0      [N,N,NTime] control
%                   maxit   [1]         maximum number of iterations
%        RETURN:    u       [N,N,NTime] optimal control

% This function implements the variable truncated inertial
% proximal method

% Initializing the global variables
global alph
global bet
global low
```

```
global up
global h
global lin
global tol_stop
global N
global NTime

% Parameters for estimating the Lipschitz constant
L=1e-4;
eta=1.5;
% Getting the inertial parameter
par2=0.5;

% Initializing
c2=1e-3;
u = u0;
it=1;
uuold=u;
stopping=1;
y=zeros(size(u));
p=y;
eps=1e-2;
L0=L;

% Calculating the gradient
if lin==0
    [y,p]=laplacesolv(-u+f,u,z,y0,eps,y,p,1);
else
    [y,p]=laplacesolv(f,u,z,y0,eps,y,p,1);
end
grad=gradient(u,y,p);

% Getting function handles for the norms using the trapez rule
x=0:1/(N-1):1;
x2=0:1/(NTime-1):1;
normL2sq=@(u)trapz(x2,trapz(x,trapz(x,u.^2)));
scalprodL2=@(u,v)trapz(x2,trapz(x,trapz(x,u.*v)));

% Starting iteration
while (stopping>tol_stop)&&(it<=maxit)

    % Estimating Lipschitz constant
    if it<2
            L=L0;
    end

```

```
    % Getting update
    utemp = u - 1./L.*grad;
    utemp = prox(utemp,L);
    utemp=max(utemp,low);
    utemp=min(utemp,up);

    % Calculating new gradient
    if lin==0
        [ytemp,ptemp]=laplacesolv(-utemp+f,utemp,z,y0,eps,y,p,1);
    else
        [ytemp,ptemp]=laplacesolv(f,utemp,z,y0,eps,y,p,1);
    end
    gradtemp=gradient(u,y,p);

    % Getting required functional values
    J2const=0.5 * normL2sq(y-z)+alph * 0.5 * normL2sq(u);
    fval = 0.5 * normL2sq(ytemp-z)+alph * 0.5 * normL2sq(utemp);
    QL=J2const+0.5*L*normL2sq(utemp-u)+scalprodL2(gradtemp,utemp-u);

    % Updating Lipschitz constant
        while (fval-QL>1e-3)
            L=eta*L;
            utemp = u - 1./L.*grad;
            utemp = prox(utemp,L);

            utemp=max(utemp,low);
            utemp=min(utemp,up);
            if lin==0
                [ytemp,ptemp]=laplacesolv(-utemp+f,utemp,z,y0,...
                                eps,ytemp,ptemp,1);
            else
                [ytemp,ptemp]=laplacesolv(f,utemp,z,y0,...
                                eps,ytemp,ptemp,1);
            end
            fval = 0.5*normL2sq(ytemp-z)+alph*0.5*normL2sq(utemp);
            QL=J2const+0.5*L*normL2sq(utemp-u)+...
                        scalprodL2(gradtemp,utemp-u);
        end

    % Getting the gradient
    grad=gradtemp;

    % Calculating the Lipschitz constant
    Lt=(L+2*c2)/1.9/(1-par2);

    % Truncation tolerance
```

```
    eps=1/(it+1)^3;

    % Updating the optimization variable
    u = u - 1./Lt.*grad+par2*(u-uuold);
    u = prox(u,Lt);
    u=max(u,low);
    u=min(u,up);
    uuold=u;

    % Getting the stopping criterion
    if lin==0
        [y,p]=laplacesolv(-u+f,u,z,y0,eps,y,p,1);
        mu=-p-alph*u;
    else
        [y,p]=laplacesolv(f,u,z,y0,eps,y,p,1);
        mu=-y.*p-alph*u;
    end
    B=u-max(0,u+mu-bet)-min(0,u+mu+bet)+max(0,u-(up)+mu-bet)+...
                min(0,u-(low)+mu+bet);
    stopping=1/sqrt(h*(length(f)-1))*norm(B(:));

    % Getting the functional value
    func=functional( u, f, z,y0 );

    % Plotting
    fprintf('it = %d stop = %.3e func = %.10e \n', ...
        it, stopping , func );

    it=it+1;
end

end

%% Proximal function
function [out] = prox( u,L )
%       INPUT:     u      [N,N,NTime] control
%                  L      [1]         Lipschitz constant
%       RETURN:    out    [N,N,NTime] proximal functional value

% This function implements the proximal functional
% to the functional \|u\|_L1

% global variable:
global bet

% Getting the absolute value:
```

```
abs_ = abs(u);

% Not dividing by zero:
abs_(abs_==0)=bet/L;

% Proximal functional:
out = max(abs_-bet/L,0).*(u./abs_);

end

%% CTIP method
function [u] = CTIP(f,z,y0,u0,L,maxit)
%         INPUT:     f       [N,N,NTime] right-hand side
%                    z       [N,N,NTime] target state
%                    y0      [N,N]       starting state
%                    u0      [N,N,NTime] control
%                    L       [1]         Lipschitz constant
%                    maxit   [1]         maximum number of iterations
%         RETURN:    u       [N,N,NTime] optimal control

% This function implements the constant truncated inertial
% proximal method

% Initializing the global variables
global alph
global bet
global low
global up
global lin
global tol_stop
global h

% Getting the inertial parameter
par2=0.5;

% Initializing
c2=1e-3;
u = u0;
uold=u;
t=1.;
it=1;
stopping=1;
y=zeros(size(u));
p=y;

% Getting the truncation tolerance
```

```
eps=2/(it+1)^3;

% Calcultating the states
if lin==0
    [y,p]=laplacesolv(-u+f,u,z,y0,eps,y,p,1);
else
    [y,p]=laplacesolv(f,u,z,y0,eps,y,p,1);
end

% Calculating the steplength 1/Lt:
Lt=(L+2*c2)/1.9/(1-par2);

% Starting iteration
while (stopping>tol_stop)&&(it<=maxit)

    % Calculating the gradient
    grad=gradient(u,y,p);

    % Getting the truncation tolerance
    eps=1/(it+1)^3;

    % Updating the optimization variable
    u = u - 1./Lt.*grad+par2*(u-uold);
    u = prox(u,Lt);
    u=max(u,low);
    u=min(u,up);
    uold=u;

    % Getting the stopping criterion
    if lin==0
        [y,p]=laplacesolv(-u+f,u,z,y0,eps,y,p,1);
        mu=-p-alph*u;
    else
        [y,p]=laplacesolv(f,u,z,y0,eps,y,p,1);
        mu=-y.*p-alph*u;
    end
    B=u-max(0,u+mu-bet)-min(0,u+mu+bet)+max(0,u-(up)+mu-bet)+...
               min(0,u-(low)+mu+bet);
    stopping=1/sqrt(h*(length(f)-1))*norm(B(:));

    % Geting the functional value
    func=functional( u, f, z,y0 );

    % Plotting
    fprintf('it␣=␣%d␣stop␣=␣%.3e␣func␣=␣%.10e␣\n', ...
        it, stopping , func );
```

```

    it=it+1;
end
end

%% Solve the Laplace problem
function [y,p] = laplacesolv(f,u,z,y0,eps,y,p,both)
%       INPUT:      f      [N,N,NTime] right-hand side
%                   u      [N,N,NTime] control
%                   z      [N,N,NTime] target state
%                   y0     [N,N]       starting state
%                   eps    [1]         truncation tolerance
%                   y      [N,N,NTime] estimation of state
%                   p      [N,N,NTime] estimation of adjoint state
%                   both   [bool]      both=1 (Calculate y and p)
%                                      both=0 (Calculate y)
%       RETURN:     y      [N,N,NTime] state
%                 p        [N,N,NTime] adjoint state

% This function implements the truncated solution of the parabolic
% Laplace Problem with conjugate gradient
% y_t-laplace y + u = f (lin=0)
% y_t-laplace y + uy= f (lin=1)
% y=0 on delta Omega

%Initializing the global variables
global lin
global NTime
global N

% Getting the time difference
dtime      = 1 / (NTime-1);

n2 = (N-1)*(N-1);                       % computing n^2

% Cutting f, z and y
f = f(2:end-1,2:end-1,:); f=reshape(f,(N-2)*(N-2),NTime);
z = z(2:end-1,2:end-1,:); z=reshape(z,(N-2)*(N-2),NTime);
y=y(2:end-1,2:end-1,:);
y(:,:,1)=y0(2:end-1,2:end-1);
y=reshape(y,(N-2)*(N-2),NTime);

% Cutting p if necessary
if both==1
    p=p(2:end-1,2:end-1,:); p=reshape(p,(N-2)*(N-2),NTime);
    p(:,NTime)=0;
```

```
end

% Set laplace:
A = (n2*gallery('poisson',N-2));

% Calculate the Inverse:
if lin==0 % linear control mechanism
    NN=(N-2)^2;
    U = 1/dtime*speye(NN,NN);
    for time=2:NTime
        func_A=@(x)(A+U)*x;
        y(:,time)=cg(func_A,f(:,time)+y(:,time-1)/dtime,eps,...
                        y(:,time-1),N*N*NTime);
    end
    if both==1
        for time=NTime-1:-1:1
            func_A=@(x)(A+U)*x;
            p(:,time)=cg(func_A,...
                        z(:,time)-y(:,time)+p(:,time+1)/dtime,...
                        eps,p(:,time+1),N*N*NTime);
        end
    end
else % bilinear control mechanism
    for time=2:NTime
        U = u(2:end-1,2:end-1,time); U = U(:);
        U = spdiags(U+1/dtime,0,length(U),length(U));
        func_A=@(x)(A+U)*x;
        y(:,time)=cg(func_A,f(:,time)+y(:,time-1)/dtime,eps,...
                        y(:,time-1),N*N);
    end
    if both==1
        for time=NTime-1:-1:1
            U = u(2:end-1,2:end-1,time); U = U(:);
            U = spdiags(U+1/dtime,0,length(U),length(U));
            func_A=@(x)(A+U)*x;
            p(:,time)=cg(func_A,...
                        z(:,time)-y(:,time)+p(:,time+1)/dtime,...
                        eps,p(:,time+1),N*N);
        end
    end
end

% Filling y
y = reshape(y,N-2,N-2,NTime);
y = horzcat(zeros([N,1,NTime]), vertcat(zeros([1,N-2,NTime]),y,...
                zeros([1,N-2,NTime])), zeros([N,1,NTime]));
```

```

% Filling p if necessary
if both==1
    p = reshape(p,N-2,N-2,NTime);
    p = horzcat(zeros([N,1,NTime]), vertcat(zeros([1,N-2,NTime]),...
                    p,zeros([1,N-2,NTime])), zeros([N,1,NTime]));
end
end

function y = laplacesolv2(f,u,y0,eps,y,ifp)
%       INPUT:      f       [N,N,NTime]  right-hand side
%                   u       [N,N,NTime]  control
%                   y0      [N,N]        starting state
%                   eps     [1]          truncation tolerance
%                   y       [N,N,NTime]  estimation of state
%                   ifp     [bool]       ifp=1 (Calculate p)
%                                        ifp=0 (Calculate y)
%       RETURN:     y       [N,N,NTime]  state

% This function implements the truncated solution of the parabolic
% Laplace Problem with gmres

%Initializing the global variables
global lin
global NTime
global N

%Getting the time difference
dtime       = 1 / (NTime-1);

n2 = (N-1)*(N-1);

%Cutting f and y or p
f = f(2:end-1,2:end-1,:); f=reshape(f,(N-2)*(N-2),NTime);
y=y(2:end-1,2:end-1,:);
if ifp==0
    y(:,:,1)=y0(2:end-1,2:end-1);
    y=reshape(y,(N-2)*(N-2),NTime);
else
    y(:,:,NTime)=0;
    y=reshape(y,(N-2)*(N-2),NTime);
end

% Set laplace:
A = (n2*gallery('poisson',N-2));

```

```
% Calculate the Inverse:
if lin==0 % linear control mechanism
    NN=(N-2)^2;
    U = 1/dtime*speye(NN,NN);
        if ifp==0
            for time=2:NTime
                func_A=@(x)(A+U)*x;
                [T,y(:,time)] = evalc('gmres(func_A,...
                                f(:,time)+y(:,time-1)/dtime,...
                                [],eps,N,[],[],y(:,time-1))');
            end
        else
            for time=NTime-1:-1:1
                func_A=@(x)(A+U)*x;
                [T,y(:,time)] = evalc('gmres(func_A,...
                                f(:,time)+y(:,time+1)/dtime,...
                                [],eps,N,[],[],y(:,time+1))');
            end
        end
else  % bilinear control mechanism
    if ifp==0
        for time=2:NTime
            U = u(2:end-1,2:end-1,time); U = U(:);
            U = spdiags(U+1/dtime,0,length(U),length(U));
            func_A=@(x)(A+U)*x;
            [T,y(:,time)] = evalc('gmres(func_A,...
                                f(:,time)+y(:,time-1)/dtime,...
                                [],eps,N,[],[],y(:,time-1))');
        end
    else
        for time=NTime-1:-1:1
            U = u(2:end-1,2:end-1,time); U = U(:);
            U = spdiags(U+1/dtime,0,length(U),length(U));
            func_A=@(x)(A+U)*x;
            [T,y(:,time)] = evalc('gmres(func_A,...
                        f(:,time)+y(:,time+1)/dtime,...
                        [],eps,N,[],[],y(:,time+1))');
        end
    end
end

% Filling y
y = reshape(y,N-2,N-2,NTime);
y = horzcat(zeros([N,1,NTime]), vertcat(zeros([1,N-2,NTime]),y,...
                        zeros([1,N-2,NTime])), zeros([N,1,NTime]));

```

```
end

%% Conjugate gradient
function [x] = cg( A,b,tol,x ,maxit)
% INPUT: A [function handle] differentiation operator
% b [N*N*NTime] right-hand side
% tol [1] truncation tolerance
% x [N*N*NTime] starting value
% maxit [1] maximum number of iterations
% RETURN: x [N,N,NTime] Solution of Ax=b

% This function implements the conjugate gradient method
% to solve Ax=b

if nargin<5
    maxit=length(b);
end

r=b-A(x);
h=r;
d=h;

for it=1:maxit
    z=A(d);
    a=r'*h/(d'*z);
    x=x+a*d;
    rold=r;
    r=r-a*z;
    hold=h;
    h=r;
    b=r'*h/(rold'*hold);
    d=r+b*d;
    if norm(r)<tol
        break
    end
end
end
```

Bibliography

[Ada75] R. A. Adams. *Sobolev Spaces*. Pure and Applied Mathematics. Academic Press, Inc, 1975.

[BA15] A. Borzì and S. G. Andrade. Second-order approximation and fast multigrid solution of parabolic bilinear optimization problems. *Advances in Computational Mathematics*, 41(2):457–488, 2015.

[BC11] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer-Verlag New York, 2011.

[Bre11] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer-Verlag New York, 2011.

[BS11] A. Borzì and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*. SIAM, Philadelphia, 2011.

[BT09] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009.

[BT11] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2011.

[CB16] G. Ciaramella and A. Borzì. A LONE code for the sparse control of quantum systems. *Computer Physics Communications*, 200:312–323, 2016.

[CCK13] E. Casas, C. Clason, and K. Kunisch. Parabolic control problems in measure spaces with sparse solutions. *SIAM Journal on Control and Optimization*, 51(1):28–63, 2013.

[CG02] C. Kanzow C. Geiger. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer Berlin Heidelberg, 2002.

[CHW12] E. Casas, R. Herzog, and G. Wachsmuth. Optimality conditions and error analysis of semilinear elliptic control problems with L^1 cost functional. *SIAM Journal on Optimization*, 22(3):795–820, 2012.

[CNQ00] X. Chen, Z. Nashed, and L. Qi. Smoothing methods and semismooth methods for nondifferentiable operator equations. *SIAM Journal on Numerical Analysis*, 38(4):1200–1216, 2000.

[CRT06a] E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

[CRT06b] E.J. Candes, J.K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[CW05] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[DE03] D. L. Donoho and M. Elad. Maximal sparsity representation via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100:2197–2202, 2003.

[DT06] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse, 2006.

[ET99] I. Ekeland and R. Témam. *Convex Analysis and Variational Problems.* Society for Industrial and Applied Mathematics, 1999.

[Eva10] L. C. Evans. *Partial Differential Equations.* American Mathematical Society, 2010.

[Fou10] S. Foucart. A note on guaranteed sparse recovery via ℓ_1-minimization. *Applied and Computational Harmonic Analysis*, 29(1):97 – 103, 2010.

[FR14] M. Fornasier and H. Rauhut. Compressive sensing. In Otmar Scherzer, editor, *Handbook of Mathematical Methods in Imaging*, pages 1–48. Springer Berlin Heidelberg, 2014.

[Gri85] P. Grisvard. *Elliptic Problems in Nonsmooth Domains.* Pitman Publishing, Boston, 1985.

[HIK02] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[HS52] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[HSW12] R. Herzog, G. Stadler, and G. Wachsmuth. Directional sparsity in optimal control of partial differential equations. *SIAM Journal on Control and Optimization*, 50(2):943–963, 2012.

[HZM10] J. Huang, S. Zhang, and D. Metaxas. Efficient MR image reconstruction for compressed MR imaging. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, volume 6361 of *Lecture Notes in Computer Science*, pages 135–142. Springer Berlin Heidelberg, 2010.

[IK03] K. Ito and K. Kunisch. Semi-smooth newton methods for state-constrained optimal control problems. *Systems & Control Letters*, 50(3):221 – 228, 2003.

[IK04] K. Ito and K. Kunisch. The primal-dual active set method for nonlinear optimal control problems with bilateral constraints. *SIAM Journal on Control and Optimization*, 43(1):357–376, 2004.

[Kac60] R. I. Kachurovskii. Monotone operators and convex functionals. *Uspekhi Mat. Nauk*, 15(4):213 – 215, 1960.

[KT09] A. Kaplan and R. Tichatschke. Proximal point method and elliptic regularization. *Nonlinear Analysis: Theory, Methods & Applications*, 71(10):4525 – 4543, 2009.

[KV09] A. Kröner and B. Vexler. A priori estimates for elliptic optimal control problems with bilinear state equation. *Journal of Computational and Applied Mathematics*, 230(2):781–802, 2009.

[LBR15] D. A. Lorenz, K. Bredies, and S. Reiterer. Minimization of non-smooth, non-convex functionals by iterative thresholding. *Journal of Optimization Theory and Applications*, 165:78–112, 2015.

[LDP07] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[Lio71] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer, Berlin, 1971.

[MQ95] J.M. Martínez and L. Qi. Inexact newton methods for solving nonsmooth equations. *Journal of Computational and Applied Mathematics*, 60(1–2):127 – 145, 1995.

[MYZC08] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty. An efficient algorithm for compressed mr imaging using total variation and wavelets. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[Nes83] Y. E. Nesterov. A method for solving the convex programming problem with covergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[Nes07] Y. E. Nesterov. Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.

[Nes13] Y. E. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[OBG+05] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[OCBP14] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial Proximal Algorithm for Nonconvex Optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[Roc76] R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[RW97] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer-Verlag, 1997.

[RWS+16] V. Ratz, T. Wech, A. Schindele, A. Sauer A. Dierks, J. Reibetanz, A Borzì, T. Bley, and H. Köstler. Dynamic 3d mr-defecography. *Submitted*, 2016.

[SB16a] A. Schindele and A. Borzì. Proximal methods for elliptic optimal control problems with sparsity cost functional. *Applied Mathematics*, 2016.

[SB16b] A. Schindele and A. Borzì. Proximal methods for parabolic optimal control problems with a sparsity promoting cost functional. *Submitted*, 2016.

[SBB+08] N. Seiberlich, F. Breuer, M. Blaimer, P. Jakob, and M. Griswold. Self-calibrating GRAPPA operator gridding for radial and spiral trajectories. *Magnetic Resonance In Medicine*, 59(4):930–935, 2008.

[SED+11] N. Seiberlich, P. Ehses, J. Duerk, R. Gilkeson, and M. Griswold. Improved radial GRAPPA calibration for real-time free-breathing cardiac imaging. *Magnetic Resonance In Medicine*, 65(2):492–505, 2011.

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[Sta09] G. Stadler. Elliptic optimal control problems with L^1-control cost and applications for the placement of control devices. *Computational Optimization and Applications*, 44(2):159–181, 2009.

[Trö09] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen. Theorie, Verfahren und Anwendungen.* Vieweg, 2009.

[Ulb11] M. Ulbrich. *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces.* SIAM, Philadelphia, 2011.

[Wil88] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, 1988.

[WKM06] D. Weishaupt, V. D. Köchli, and B. Marincek. *How Does MRI Work?* Springer Berlin Heidelberg, 2006.

[WMG00] D. O. Walsh, M. W. Marcellin, and A. F. Gmitro. Adaptive reconstruction and enhancement of phased array MR imagery, 2000.

[WSS+16] T. Wech, N. Seiberlich, A. Schindele, V. Grau, L. Diffley, M. L. Gyngell, A. Borzì, H. Köstler, and J. E. Schneider. Development of real-time magnetic resonance imaging of mouse hearts at 9.4 Tesla – simulations and first application. *IEEE Transactions on Medical Imaging*, 35(3):912–920, 2016.

[WW10] G. Wachsmuth and D. Wachsmuth. Convergence and regularisation results for optimal control problems with sparsity function. *ESAIM Control Optimisation and Calculus of Variations*, 17(3):858–886, 2010.