Identification of sparsely representable diffusion parameters in elliptic problems

Luzia N. Felber, Helmut Harbrecht, Marc Schmidlin

Departement Mathematik und Informatik
Fachbereich Mathematik
Universität Basel
CH-4051 Basel

Preprint No. 2023-03
April 2023

dmi.unibas.ch

Identification of sparsely representable diffusion parameters in elliptic problems

Luzia N. Felber*, Helmut Harbrecht*, and Marc Schmidlin*

Abstract. We consider the task of estimating the unknown diffusion parameter in an elliptic PDE as a model problem to develop and test the effectiveness and robustness to noise of reconstruction schemes with sparsity regularisation. To this end, the model problem is recasted as a nonlinear optimal control problem, where the unknown diffusion parameter is modelled using a linear combination of the elements of a known bounded sequence of functions with unknown coefficients. We show that the regularisation of this nonlinear optimal control problem using a weighted ℓ^1-norm has minimisers that are finitely supported. We then propose modifications of well-known algorithms (ISTA and FISTA) to find a minimiser of this weighted ℓ^1-norm regularised nonlinear optimal control problem that account for the fact that in general the coefficients need to be ℓ^1 and not only ℓ^2 summable. We also introduce semismooth methods (ASISTA and FASISTA) for finding a minimiser, which locally use Gauss-Newton type surrogate models that additionally are stabilised by means of a Levenberg-Marquardt type approach. Our numerical examples show that the regularisation with the weighted ℓ^1-norm indeed does make the estimation more robust with respect to noise. Moreover, the numerical examples also demonstrate that the ASISTA and FASISTA methods are quite efficient, outperforming both ISTA and FISTA.

Key words. Parameter identification, nonlinear optimal control, ℓ^1-regularisation, iterated soft-thresholding, semismooth method, Levenberg-Marquardt method

MSC codes. 49M05, 49M15, 65N21

1. Introduction. In many applications, one has a physical phenomenon that is described by a partial differential equation (PDE), where one is able to obtain certain measurements and wants to reconstruct other involved quantities. In a mathematical context, such problems are commonly called inverse or parameter estimation problems.

For example, magnetic resonance elastography (MRE) is becoming more prevalent in clinical diagnostic as it is a powerful tool to map tissue stiffness. As a noninvasive technique, it is currently well established to examine the liver, but it can also be used to diagnose breast cancer, to study the function of the heart or to monitor mechanical muscle properties [12]. Further applications include imaging the brain to diagnose early stages of Alzheimer's disease as well as determine its progress [23].

To obtain an MRE, a stress or motion is applied to the tissue under consideration, the response of which is then measured by magnetic resonance imaging (MRI). This data and the unknown stiffness parameter are related by a viscoelastic wave equation, which leads to a generalisation of the Helmholtz equation, when the motion or stress applied to the tissue is periodic. Using this PDE, inversion algorithms can reconstruct the stiffness parameter to generate the elastogram of the mechanical properties [12].

Current research in biomedical engineering investigates the possibility to reduce the magnetic field in MRIs which would be beneficial in many practical applications. For example, this enables the construction of mobile apparatures, and the lower power requirements has less

[*] Departement Mathematik und Informatik, Universität Basel, Spiegelgasse 1, 4051 Basel, Schweiz (luzia.felber@unibas.ch, helmut.harbrecht@unibas.ch, marc.schmidlin@unibas.ch).

environmental impact. However, the reduced magnetic field yields noisier measurements and thus also noisier MRI images [26]. Therefore, there is a need for robust inversion algorithms to compute elastograms in this setting.

In this article, we investigate methods for the identification of a parameter in a simpler model problem: the diffusion parameter function in a second-order diffusion model. In particular, we consider an approach that represents the parameter function sought using a linear combination of the elements of a known bounded sequence of functions with unknown coefficients, i.e. an expansion. This enables us to formulate the inversion as a nonlinear optimal control problem, where we are then minimising a functional that depends on the coefficients of the expansion. To regularise this minimisation task, we additionally add a weighted ℓ^1-norm of the coefficients to the functional.

As the space, in which the parameter lies, is not a Hilbert space, and since we only assume that the sequence of elements, which is utilised in the expansion, is bounded, we require that the coefficients form an ℓ^1-sequence. Therefore, in order to justify the use of the well-established iterative shrinkage-thresholding algorithm (ISTA, see [8]) and the fast iterative-shrinkage thresholding algorithm (FISTA, see [2]) for the minimisation, we show that the soft-threshold based first order optimality condition, which lies at the center of these two methods, also holds in our setting. Another popular approach to solve the underlying optimisation problem is given by the alternating direction method of multipliers (ADMM, see [3]), which we however do not consider here.

For the optimisation, we also introduce an active set method similar to those proposed by several authors, compare [13, 19, 21, 22] for example, which we call the active set iterated soft-threshold algorithm (ASISTA). However, in contrast to the active set methods cited for nonlinear optimal control problems, the ASISTA method is based on the semismooth minimisation of successive Gauss-Newton type approximations of the functional, which are additionally stabilised by using a type of Levenberg-Marquardt stabilisation. In order to derive this method, we also provide the semismoothness of the soft-threshold based first order optimality condition for our setting, as this setting is not covered by the works cited. Moreover, we also introduce the fast active set iterated soft-threshold algorithm (FASISTA) by simply applying the acceleration from [24] to the ASISTA method.

We finally discretise our model problem using bilinear finite elements and consider two expansions: one which represents the unknown diffusion parameter using Haar wavelets and one that is based on the discrete cosine transform. With this we test the robustness of our approach in numerical experiments by using different regularisation parameters and noise levels and compare the behaviour of the three optimisation methods used. It turns out our new ASISTA and FASISTA methods converge at a higher rate compared to the other two methods, hence being superior.

This article is structured as follows. In Section 2, we introduce the optimal control problem under consideration. Then, in Section 3, we compute the cost functional's derivative and derive the first order optimality condition. The optimisation algorithms which we apply are proposed in Section 4. The discretisation of the optimal control problem is introduced in Section 5. Section 6 contains the results of our numerical experiments. Finally, in Section 7, we state concluding remarks.

2. Parameter identification problem. As the model problem, we consider the following second-order elliptic PDE on the domain $\Omega \subset \mathbb{R}^n$ with boundary $\Gamma = \partial\Omega$,

$$-\operatorname{div}(a\nabla u) = f \text{ in } \Omega, \quad u = g \text{ on } \Gamma. \tag{2.1}$$

Here, the source term $f \in H^{-1}(\Omega)$ and the boundary values $g \in H^{1/2}(\Gamma)$ are assumed to be known input data, while the diffusion parameter function

$$a \in A_{\mathrm{ad}} := \Big\{ v \in L^\infty(\Omega) : \operatorname*{ess\,inf}_{\boldsymbol{x}\in\Omega} v(\boldsymbol{x}) > 0 \Big\} \subset L^\infty(\Omega)$$

is not known. However, we assume that $u \in H^1(\Omega)$ can be measured yielding the measurement $u_d \in L^2(\Omega)$, which, due to noise in the measuring procedure, only fulfils $\|u - u_d\|_{L^2(\Omega)} \approx 0$. Then, the parameter identification problem is to determine the unknown diffusion parameter function $a \in A_{\mathrm{ad}}$.

Because of the fact that u_d only is in $L^2(\Omega)$, one cannot simply replace u in (2.1) with u_d to arrive at a nonlinear operator equation to be solved. Instead, it is common to reformulate the problem as a constrained minimisation, yielding the nonlinear optimal control problem:

$$\begin{aligned} \text{minimise} \quad & \frac{1}{2}\|u - u_d\|^2_{L^2(\Omega)} \quad \text{over } a \in A_{\mathrm{ad}},\ u \in H^1(\Omega), \\ \text{subject to} \quad & -\operatorname{div}(a\nabla u) = f \text{ in } \Omega, \quad u = g \text{ on } \Gamma. \end{aligned}$$

Using the *parameter-to-state mapping* $S\colon A_{\mathrm{ad}} \to H^1(\Omega)$, that is the map $S(a) = u$ stemming from (2.1), we arrive at the equivalent reduced formulation:

$$\text{minimise} \quad \frac{1}{2}\big\|S(a) - u_d\big\|^2_{L^2(\Omega)} \quad \text{over } a \in A_{\mathrm{ad}}.$$

Since it is well established that this problem and other similar reformulations are ill-posed, see e.g. [1, 12, 26], it is necessary to introduce more knowledge of the possible or likely diffusion parameter function $a \in A_{\mathrm{ad}}$ into the formulation, see [10, 16] for example.

We propose to consider the situation, where it is known or assumed that the logarithm of the diffusion parameter function $a \in A_{\mathrm{ad}}$, which is to be reconstructed, can be approximated by a sparse linear combination of the elements of a known bounded sequence $\boldsymbol{\psi} = (\psi_k)_{k\in\Lambda} \subset L^\infty(\Omega)$, where the index set Λ is countable but may be finite or infinite. That is, we assume that we have

$$\log(a) \approx \sum_{k\in\Lambda} b_k \psi_k$$

for some sparse sequence $\boldsymbol{b} = (b_k)_{k\in\Lambda} \in \mathbb{R}^\Lambda$.

For this, we first introduce the sequence spaces ℓ^p with $1 \le p < \infty$ and ℓ^∞ by

$$\begin{aligned} \ell^p &:= \Big\{ \boldsymbol{v} \in \mathbb{R}^\Lambda : \sum_{k\in\Lambda} |v_k|^p < \infty \Big\}, & \|\boldsymbol{v}\|_{\ell^p} &:= \Big(\sum_{k\in\Lambda} |v_k|^p \Big)^{1/p}, \\ \ell^\infty &:= \Big\{ \boldsymbol{v} \in \mathbb{R}^\Lambda : \max_{k\in\Lambda} |v_k| < \infty \Big\}, & \|\boldsymbol{v}\|_{\ell^\infty} &:= \max_{k\in\Lambda} |v_k| \end{aligned}$$

and the $\boldsymbol{y}$-weighted sequence spaces $\ell^p_{\boldsymbol{y}}$ with $1 \leq p < \infty$ and $\ell^\infty_{\boldsymbol{y}}$ by

$$\ell^p_{\boldsymbol{y}} := \Big\{ \boldsymbol{v} \in \mathbb{R}^\Lambda : \sum_{k\in\Lambda} |y_k v_k|^p < \infty \Big\}, \qquad \|\boldsymbol{v}\|_{\ell^p_{\boldsymbol{y}}} := \Big(\sum_{k\in\Lambda} |y_k v_k|^p \Big)^{1/p},$$
$$\ell^\infty_{\boldsymbol{y}} := \Big\{ \boldsymbol{v} \in \mathbb{R}^\Lambda : \max_{k\in\Lambda} |y_k v_k| < \infty \Big\}, \qquad \|\boldsymbol{v}\|_{\ell^\infty_{\boldsymbol{y}}} := \max_{k\in\Lambda} |y_k v_k|$$

for any $\boldsymbol{y} \in \mathbb{R}^\Lambda_{>0}$. With these at hand, we define the expansion mapping $E\colon \ell^1 \to L^\infty(\Omega)$ by

$$E(\boldsymbol{b}) := \sum_{k\in\Lambda} b_k \psi_k. \tag{2.2}$$

Hence, we are proposing to search for the diffusion parameter function in the subspace

$$\Big\{ \exp\big(E(\boldsymbol{b})\big) : \boldsymbol{b} \in \ell^1 \Big\} \subset A_{\mathrm{ad}}$$

and thus define the *data misfit mapping* $M\colon \ell^1 \to L^2(\Omega)$ and the *data fidelity functional* $\mathcal{F}\colon \ell^1 \to \mathbb{R}$ by

$$M(\boldsymbol{b}) := S\Big(\exp\big(E(\boldsymbol{b})\big)\Big) - u_d \quad \text{and} \quad \mathcal{F}(\boldsymbol{b}) := \frac{1}{2} \big\| M(\boldsymbol{b}) \big\|^2_{L^2(\Omega)}. \tag{2.3}$$

The corresponding optimal control problem thus now simply reads:

$$\text{minimise} \quad \mathcal{F}(\boldsymbol{b}) \quad \text{over } \boldsymbol{b} \in \ell^1.$$

Now it is known, at least if we had the space ℓ^2 instead of ℓ^1, cf. [8, 13], that to encourage sparsity one may introduce the *$\boldsymbol{w}$-weighted ℓ^1-regularisation term* $\mathcal{R}\colon \ell^1_{\boldsymbol{\mu}} \to \mathbb{R}$ defined by

$$\mathcal{R}(\boldsymbol{b}) := \sum_{k\in\Lambda} w_k |b_k|, \tag{2.4}$$

where $\boldsymbol{w} \in \mathbb{R}^\Lambda_{\geq 0}$ is a non-negative sequence. In order for (2.4) to be welldefined, we assume that the positive sequence $\boldsymbol{\mu} \in \mathbb{R}^\Lambda_{>0}$ is such that

$$\mu_k \geq \underline{\mu} \quad \text{and} \quad \mu_k \geq w_k$$

holds for all $k \in \Lambda$ for some $\underline{\mu} \in \mathbb{R}_{>0}$. Additionally, when Λ is an infinite set, we assume that $\boldsymbol{w}$ and therefore also $\boldsymbol{\mu}$ tend to infinity. With this, we finally arrive at the regularised minimisation problem

$$\text{minimise} \quad J(\boldsymbol{b}) := \frac{1}{2} \big\| M(\boldsymbol{b}) \big\|^2_{L^2(\Omega)} + \mathcal{R}(\boldsymbol{b}) \quad \text{over} \quad \boldsymbol{b} \in \ell^1_{\boldsymbol{\mu}}. \tag{2.5}$$

Note that for the rest of the article, we choose to rig the ℓ^p-sequence spaces and the $\boldsymbol{y}$-weighted ℓ^p-sequence spaces over Λ around the Hilbert space ℓ^2 with its associated scalar product

$$\langle \boldsymbol{d}, \boldsymbol{b} \rangle := \sum_{k\in\Lambda} d_k b_k.$$

This then yields the following schema of spaces with canonical embeddings and identifications of duals under the duality product $\langle\cdot,\cdot\rangle$, where $1 < p, q < \infty$ are conjugate indices, that is $1/p + 1/q = 1$:

$$\begin{tikzcd}
 & \ell^q \arrow[dr, hook] & \\
\ell^1_{\boldsymbol{y}} \arrow[ur, hook] \arrow[dr, hook] \arrow[dd, hook] & & \ell^\infty \arrow[dd, hook] \\
 & \ell^2 \arrow[dr, hook] & \\
\ell^1 \arrow[dr, hook] & & \ell^\infty_{\boldsymbol{y}^{-1}} \\
 & \ell^p \arrow[ur, hook] &
\end{tikzcd}$$

$$\begin{aligned}
(\ell^1_{\boldsymbol{y}})^* &= \ell^\infty_{\boldsymbol{y}^{-1}} \\
(\ell^1)^* &= \ell^\infty \\
(\ell^p)^* &= \ell^q \\
(\ell^2)^* &= \ell^2 \\
(\ell^q)^* &= \ell^p
\end{aligned}$$

Especially, the dual of a space on the left side is thus identified with the space lying diagonally opposite it. Lastly, for any $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{R}^\Lambda$, we define $\boldsymbol{v} \cdot \boldsymbol{u} := (v_k u_k)_{k\in\Lambda}$ and we set $\boldsymbol{v}^{-1} := (v_k^{-1})_{k\in\Lambda}$ for any $\boldsymbol{v} \in \mathbb{R}^\Lambda_{\neq 0}$.

3. Derivatives and first order optimality conditions.

3.1. Derivatives of the data misfit and data fidelity terms. We shall next consider the behaviour of the data misfit mapping M and the data fidelity functional $\mathcal{F}$. To this end, we show the following lemma which provides the Fréchet derivatives of the problem under consideration.

Lemma 3.1. *Both $M\colon \ell^1 \to L^2(\Omega)$ and $\mathcal{F}\colon \ell^1 \to \mathbb{R}$ are Fréchet differentiable and their derivatives are given by*

$$M'(\boldsymbol{b})[\boldsymbol{d}] = u' \quad \text{and} \quad \mathcal{F}'(\boldsymbol{b})[\boldsymbol{d}] = -\big(aE(\boldsymbol{d})\nabla u, \nabla p\big)_{L^2(\Omega)},$$

where, with $a = \exp\big(E(\boldsymbol{b})\big)$ and $u = S(a)$, that is u solves

$$-\operatorname{div}(a\nabla u) = f \text{ in } \Omega, \quad u = g \text{ on } \Gamma,$$

$u' \in H^1_0(\Omega)$ is the solution of the boundary value problem

$$-\operatorname{div}(a\nabla u') = \operatorname{div}\big(aE(\boldsymbol{d})\nabla u\big) \text{ in } \Omega, \quad u' = 0 \text{ on } \Gamma,$$

and the adjoint state $p \in H^1_0(\Omega)$ satisfies the boundary value problem

$$-\operatorname{div}(a\nabla p) = u - u_d \text{ in } \Omega, \quad p = 0 \text{ on } \Gamma.$$

Proof. In accordance with e.g. [1, 15], S is Fréchet differentiable and $S'(a)$ is given by $S'(a)[v] = u'$, where u' is the solution of the boundary value problem

$$-\operatorname{div}(a\nabla u') = \operatorname{div}(v\nabla u) \text{ in } \Omega, \quad u' = 0 \text{ on } \Gamma,$$

with $u = S(a)$. Moreover, E is a bounded linear map and, hence, also Fréchet differentiable with $E'(\boldsymbol{b})[\boldsymbol{v}] = E(\boldsymbol{v})$, while

$$\exp\colon L^\infty(\Omega) \to L^\infty(\Omega), \quad a \mapsto \sum_{j=0}^\infty \frac{a^j}{j!},$$

being a globally converging power series on the Banach algebra $L^\infty(\Omega)$, is Fréchet differentiable with

$$\exp'(a)[v] = \sum_{j=1}^{\infty} \frac{a^{j-1} j v}{j!} = \sum_{j=0}^{\infty} \frac{a^j}{j!} v = \exp(a)v.$$

Now simply applying the chain rule for Fréchet derivatives on $M = S \circ \exp \circ E$ yields the assertions for M.

The Fréchet differentiability of $\mathcal{F}$ is again a consequence of the chain rule. Using it and the adjoint state p, we arrive at

$$\mathcal{F}'(\boldsymbol{b})[\boldsymbol{d}] = \big(u - u_d, M'(\boldsymbol{b})[\boldsymbol{d}]\big)_{L^2(\Omega)} = (u - u_d, u')_{L^2(\Omega)} = -\big(\operatorname{div}(a\nabla p), u'\big)_{L^2(\Omega)}.$$

The formula for $\mathcal{F}'(\boldsymbol{b})$ then obviously follows by integration by parts,

$$\begin{aligned} -\big(\operatorname{div}(a\nabla p), u'\big)_{L^2(\Omega)} &= -\big(p, \operatorname{div}(a\nabla u')\big)_{L^2(\Omega)} \\ &= \Big(p, \operatorname{div}\big(aE(\boldsymbol{d})\nabla u\big)\Big)_{L^2(\Omega)} = -\big(aE(\boldsymbol{d})\nabla u, \nabla p\big)_{L^2(\Omega)}. \end{aligned}$$ ■

Remark 3.2. It is well known that the parameter-to-state mapping $S\colon A_{\mathrm{ad}} \to H^1(\Omega)$ is a real analytic mapping, see e.g. [7, Section 2.1]. Therefore, as the mappings exp and E are obviously also real analytic, it follows by the chain rule for analytic mappings that both M and $\mathcal{F}$ are real analytic mappings and thus indeed infinitely Fréchet differentiable.

3.2. Generalised derivative of the regularisation term.

In order to derive a first order necessary condition for any minimiser of J, we now consider the regularisation term. Since $\mathcal{R}\colon \ell^1_{\boldsymbol{\mu}} \to \mathbb{R}$ is obviously locally Lipschitz, it is *generalised differentiable* everywhere, cf. [6, Proposition 2.1.2], and its generalised derivative is characterised as follows.

Lemma 3.3. *The functional $\mathcal{R}\colon \ell^1_{\boldsymbol{\mu}} \to \mathbb{R}$ is generalised differentiable and its generalised derivative is given by*

$$\partial\mathcal{R}(\boldsymbol{b}) = \big\{\boldsymbol{\xi}_{\boldsymbol{\theta}} \in \ell^\infty_{\boldsymbol{\mu}^{-1}} : \boldsymbol{\theta} \in \Theta(\boldsymbol{b})\big\} \quad \textit{with} \quad \boldsymbol{\xi}_{\boldsymbol{\theta}} = (\theta_k w_k)_{k\in\Lambda},$$

where

$$\Theta(\boldsymbol{b}) := \big\{\boldsymbol{\theta} \in [-1,1]^\Lambda : \theta_k = 1 \textit{ if } b_k > 0 \textit{ and } \theta_k = -1 \textit{ if } b_k < 0 \textit{ for all } k \in \Lambda\big\}.$$

Note that we have used the identification of the dual $(\ell^1_{\boldsymbol{\mu}})^* = \ell^\infty_{\boldsymbol{\mu}^{-1}}$ here, so that a $\boldsymbol{\xi}_{\boldsymbol{\theta}}$ is indeed representing a linear functional under $\langle\cdot,\cdot\rangle$,

$$\langle \boldsymbol{\xi}_{\boldsymbol{\theta}}, \boldsymbol{d}\rangle = \sum_{k\in\Lambda} \theta_k w_k d_k.$$

Proof. As $\mathcal{R}\colon \ell^1_{\boldsymbol{\mu}} \to \mathbb{R}$ is not only locally Lipschitz but also convex, we know that $\mathcal{R}$ has a subderivative everywhere. This is equal to the generalised derivative. We also know that the generalised directional derivative $\mathcal{R}^\circ(b;d)$ simply equals the directional derivative $\mathcal{R}'(b;d)$, see [6, Proposition 2.2.7],

$$\mathcal{R}^\circ(\boldsymbol{b};\boldsymbol{d}) := \limsup_{\boldsymbol{y}\to\boldsymbol{b},\,\varepsilon\downarrow 0} \frac{\mathcal{R}(\boldsymbol{y}+\varepsilon\boldsymbol{d})-\mathcal{R}(\boldsymbol{y})}{\varepsilon} = \lim_{\varepsilon\downarrow 0} \frac{\mathcal{R}(\boldsymbol{b}+\varepsilon\boldsymbol{d})-\mathcal{R}(\boldsymbol{b})}{\varepsilon} =: \mathcal{R}'(\boldsymbol{b};\boldsymbol{d}).$$

With this we have

$$\mathcal{R}^\circ(\boldsymbol{b};\boldsymbol{d}) = \lim_{\varepsilon\downarrow 0} \sum_{k\in\Lambda} w_k \frac{|b_k+\varepsilon d_k|-|b_k|}{\varepsilon} \le \sum_{k\in\Lambda} w_k|d_k| \le \|d\|_{\ell^1_\mu}.$$

Next, we will prove

$$\lim_{\varepsilon\downarrow 0} \sum_{k\in\Lambda,\,b_k\neq 0} w_k \frac{|b_k+\varepsilon d_k|-|b_k|}{\varepsilon} = \sum_{k\in\Lambda,\,b_k>0} w_k d_k - \sum_{k\in\Lambda,\,b_k<0} w_k d_k.$$

To that end, let $(\varepsilon_j)_{j\in\mathbb{N}}\subset\mathbb{R}$ be an arbitrary sequence fulfilling $\varepsilon_j\downarrow 0$. We fix an arbitrary $\delta>0$ and can then find a finite set $\Lambda_\delta\subset\Lambda$ such that

$$\sum_{k\in\Lambda\setminus\Lambda_\delta} \mu_k|d_k| \le \frac{\delta}{2},$$

which we use to introduce $m_\delta := \min\{|b_k| : k\in\Lambda_\delta \text{ with } b_k\neq 0\}$. Clearly, we have $m_\delta>0$ and, therefore, there is a $j_\delta\in\mathbb{N}$ such that

$$\varepsilon_j\|d\|_{\ell^1_\mu} \le \underline{\mu}\frac{m_\delta}{2}$$

holds for all $j\ge j_\delta$. Thus, for all $j\ge j_\delta$ and all $k\in\Lambda_\delta$ with $b_k\neq 0$, we have

$$\varepsilon_j|d_k| \le \frac{\mu_k}{\underline{\mu}}\varepsilon_j d_k \le \frac{1}{\underline{\mu}}\varepsilon_j\|d\|_{\ell^1_\mu} \le \frac{m_\delta}{2},$$

which implies that $b_k+\varepsilon_j d_k$ has the same sign as b_k. Consequently, for all $j\ge j_\delta$, we have

$$\sum_{k\in\Lambda_\delta,\,b_k\neq 0} w_k \frac{|b_k+\varepsilon_j d_k|-|b_k|}{\varepsilon_j} = \sum_{k\in\Lambda_\delta,\,b_k>0} w_k d_k - \sum_{k\in\Lambda_\delta,\,b_k<0} w_k d_k$$

and we arrive at

$$\left|\sum_{k\in\Lambda,\,b_k\neq 0} w_k \frac{|b_k+\varepsilon_j d_k|-|b_k|}{\varepsilon} - \sum_{k\in\Lambda,\,b_k>0} w_k d_k + \sum_{k\in\Lambda,\,b_k<0} w_k d_k\right| \le 2 \sum_{k\in\Lambda\setminus\Lambda_\delta,\,b_k\neq 0} w_k|d_k|.$$

As we have

$$2\sum_{k\in\Lambda\setminus\Lambda_\delta,\,b_k\neq 0} w_k|d_k| \le 2\sum_{k\in\Lambda\setminus\Lambda_\delta} w_k|d_k| \le 2\sum_{k\in\Lambda\setminus\Lambda_\delta} \mu_k|d_k| \le \delta$$

and $\delta > 0$ was arbitrary, this shows

$$\lim_{j\to\infty} \sum_{k\in\Lambda,\, b_k\neq 0} w_k \frac{|b_k+\varepsilon_j d_k| - |b_k|}{\varepsilon_j} = \sum_{k\in\Lambda,\, b_k>0} w_k d_k - \sum_{k\in\Lambda,\, b_k<0} w_k d_k.$$

Lastly, it is immediately evident that

$$\lim_{\varepsilon\downarrow 0} \sum_{k\in\Lambda,\, b_k=0} w_k \frac{|b_k+\varepsilon d_k| - |b_k|}{\varepsilon_j} = \lim_{\varepsilon\downarrow 0} \sum_{k\in\Lambda,\, b_k=0} w_k \frac{\varepsilon |d_k|}{\varepsilon_j} = \sum_{k\in\Lambda,\, b_k=0} w_k |d_k|,$$

which proves

$$\mathcal{R}^\circ(\boldsymbol{b};\boldsymbol{d}) = \sum_{k\in\Lambda,\, b_k>0} w_k d_k - \sum_{k\in\Lambda,\, b_k<0} w_k d_k + \sum_{k\in\Lambda,\, b_k=0} w_k |d_k|.$$

Finally, let $\boldsymbol{\xi} \in \ell^\infty_{\boldsymbol{\mu}^{-1}}$ fulfil $\mathcal{R}^\circ(\boldsymbol{b};\boldsymbol{d}) \geq \langle \boldsymbol{\xi}, \boldsymbol{d}\rangle$ for all $\boldsymbol{d} \in \ell^1_{\mu}$. Now, we introduce the sequences $\boldsymbol{e}^{(j)} \in \ell^1_{\boldsymbol{\mu}}$ for $j \in \Lambda$ defined by $e^{(j)}_k = \delta_{j,k}$. Obviously, we have

$$\xi_k = \langle \boldsymbol{\xi}, \boldsymbol{e}^{(k)}\rangle \leq \mathcal{R}^\circ(\boldsymbol{b};\boldsymbol{e}^{(k)}) \quad \text{and} \quad \xi_k = -\langle \boldsymbol{\xi}, -\boldsymbol{e}^{(k)}\rangle \geq -\mathcal{R}^\circ(\boldsymbol{b};-\boldsymbol{e}^{(k)})$$

for every $k \in \Lambda$. If $b_k > 0$ this yields $\xi_k = w_k$ and, similarily, $\xi_k = -w_k$, when $b_k < 0$. For $b_k = 0$ we simply get $|\xi_k| \leq w_k$. Hence, there is a $\boldsymbol{\theta} \in \Theta(\boldsymbol{b})$ such that $\boldsymbol{\xi} = \boldsymbol{\xi}_{\boldsymbol{\theta}}$. Conversely, since for every $\boldsymbol{\theta} \in \Theta(\boldsymbol{b})$ we clearly have $\mathcal{R}^\circ(\boldsymbol{b};\boldsymbol{d}) \geq \langle \boldsymbol{\xi}_{\boldsymbol{\theta}}, \boldsymbol{d}\rangle$ for all $\boldsymbol{d} \in \ell^1_{\boldsymbol{\mu}}$, it follows that

$$\partial\mathcal{R}(\boldsymbol{b}) = \big\{\boldsymbol{\xi}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta(\boldsymbol{b})\big\}. \qquad \blacksquare$$

3.3. The first order optimality condition. Using the results of the previous two subsections, it follows that $J\colon \ell^1_{\boldsymbol{\mu}} \to \mathbb{R}$ is generalised differentiable everywhere, cf. [6, Proposition 2.3.3], and this implies a necessary first order condition for any local minimiser of J, see [6, Proposition 2.3.2]. Moreover, the formula for the generalised derivative follows by [6, Corollary 1 of Proposition 2.3.3].

Proposition 3.4. *The generalised derivative of J is given by*

$$\partial J(\boldsymbol{b}) = \mathcal{F}'(\boldsymbol{b}) + \partial\mathcal{R}(\boldsymbol{b}) \subset \ell^\infty_{\boldsymbol{\mu}^{-1}}.$$

Moreover, any local minimiser $\boldsymbol{b}^\star$ of J must fulfil $\mathbf{0} \in \partial J(\boldsymbol{b}^\star) \subset \ell^\infty_{\boldsymbol{\mu}^{-1}}$.

Considering a local minimiser $\boldsymbol{b}^\star$ of J, we set $\boldsymbol{g}^\star := \mathcal{F}'(\boldsymbol{b}^\star) \in \ell^\infty \subset \ell^\infty_{\boldsymbol{\mu}^{-1}}$. Then, we have to have $\boldsymbol{g}^\star + \boldsymbol{\xi}_{\boldsymbol{\theta}} = 0$ for some $\boldsymbol{\theta} \in \Theta(\boldsymbol{b}^\star)$, which we can also state as

$$\begin{cases} g^\star_k = -w_k, & \text{if } b^\star_k > 0,\\ g^\star_k = w_k, & \text{if } b^\star_k < 0,\\ |g^\star_k| \leq w_k, & \text{if } b^\star_k = 0.\end{cases} \tag{3.1}$$

Since we know that the terms $g^\star_k$ are bounded while the terms w_k tend to infinity, it follows that the first two cases, and hence $b^\star_k \neq 0$, can only occur for finitely many k. Thus, we can conclude what is already known to be true for the Hilbert space setting: any minimiser of (2.5) is *sparse* in the sense that it is a *finitely supported* sequence.

Nonetheless, this first order optimality condition is not well-suited for numerical exploitation. Therefore, we proceed to show that the soft-threshold based first order optimality condition used in a Hilbert space setting, i.e. for $\mathcal{F}\colon \ell^2 \to \mathbb{R}$, see [8, 13] for example, applies to our non-reflexive, non-smooth Banach space setting with $\mathcal{F}\colon \ell^1 \to \mathbb{R}$ also.

To this end, we introduce the soft-threshold operator $\boldsymbol{T}_{\boldsymbol{w}}\colon \mathbb{R}^\Lambda \to \mathbb{R}^\Lambda$ by

$$\boldsymbol{T}_{\boldsymbol{w}}(\boldsymbol{b}) := \Big(\operatorname{sgn}(b_k)\max\big\{0, |b_k| - w_k\big\}\Big)_{k\in\Lambda}. \tag{3.2}$$

Then, as is already known in the Hilbert space setting, we have the following equivalence.

Theorem 3.5. *The first order optimality condition* (3.1) *is equivalent to*

$$\boldsymbol{b}^\star = \boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}\big(\boldsymbol{b}^\star - \boldsymbol{s}\cdot\mathcal{F}'(\boldsymbol{b}^\star)\big) \tag{3.3}$$

where $\boldsymbol{s} \in \mathbb{R}^\Lambda_{>0}$ *is any arbitrarily chosen positive sequence. Moreover, the first order optimality condition* (3.1) *implies* (3.3) *for any arbitrarily chosen non-negative sequence* $\boldsymbol{s} \in \mathbb{R}^\Lambda_{\geq 0}$.

Proof. Let $\boldsymbol{g}^\star := \mathcal{F}'(\boldsymbol{b}^\star)$. We first assume that condition (3.1) holds. Then, the elements of the sequence $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{b}^\star - \boldsymbol{s}\cdot\boldsymbol{g}^\star)$ are given by

$$\begin{aligned}
&\operatorname{sgn}(b_k^\star - s_k g_k^\star)\max\Big\{0, |b_k^\star - s_k g_k^\star| - s_k w_k\Big\}\\
&\quad= \begin{cases}
\operatorname{sgn}(b_k^\star + s_k w_k)\max\big\{0, |b_k^\star + s_k w_k| - s_k w_k\big\} = b_k^\star, & \text{when } b_k^\star > 0,\\
\operatorname{sgn}(b_k^\star - s_k w_k)\max\big\{0, |b_k^\star - s_k w_k| - s_k w_k\big\} = b_k^\star, & \text{when } b_k^\star < 0,\\
\operatorname{sgn}(-s_k g_k^\star)\max\big\{0, |s_k g_k^\star| - s_k w_k\big\} = 0 = b_k^\star, & \text{when } b_k^\star = 0,
\end{cases}
\end{aligned}$$

when $s_k \geq 0$ holds for all $k \in \Lambda$. This proves that condition (3.3) is fulfilled.

Now, let us assume that condition (3.3) holds for an arbitrarily chosen positive sequence $\boldsymbol{s} \in \mathbb{R}^\Lambda_{>0}$. Then, $\boldsymbol{b}^\star = \boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{b}^\star - \boldsymbol{s}\cdot\boldsymbol{g}^\star)$ and we have

$$b_k^\star = \operatorname{sgn}(b_k^\star - s_k g_k^\star)\max\big\{0, |b_k^\star - s_k g_k^\star| - s_k w_k\big\}$$

for all $k \in \Lambda$. If $b_k^\star > 0$, we necessarily have

$$b_k^\star - s_k g_k^\star > 0 \quad \text{and} \quad b_k^\star = |b_k^\star - s_k g_k^\star| - s_k w_k$$

as the sign-term must be positive and the max-term cannot equal 0, respectively. However, this implies

$$b_k^\star = b_k^\star - s_k g_k^\star - s_k w_k \quad \text{or equivalently} \quad g_k^\star = -w_k.$$

Mutatis mutandis, when $b_k^\star < 0$, we arrive at

$$b_k^\star = b_k^\star - s_k g_k^\star + s_k w_k \quad \text{or equivalently} \quad g_k^\star = w_k.$$

Finally, if $b_k^\star = 0$, we have

$$0 = \operatorname{sgn}(-s_k g_k^\star) \max\bigl\{0, |s_k g_k^\star| - s_k w_k\bigr\}$$

which implies

$$|s_k g_k^\star| - s_k w_k \leq 0 \quad \text{or equivalently} \quad |g_k^\star| \leq w_k.$$

Hence, we have that condition (3.1) is fulfilled. ■

It is informative to consider in which spaces the terms in the right-hand side of the first order optimality condition (3.3) lie. For this, we will restrict the possible choices of the step size parameter $\boldsymbol{s} \in \mathbb{R}_{\geq 0}^{\Lambda}$ slightly: We assume that there is a $c \in \mathbb{R}_{>0}$ such that

$$s_k \geq \frac{c}{\mu_k} \tag{3.4}$$

holds for all $k \in \Lambda$. Note that when Λ is a finite index set this simply means that $s_k > 0$ holds for all $k \in \Lambda$, however, when Λ is an infinite index set, then μ_k tends to infinity and (3.4) simply means that we are requiring that s_k does not tend to zero faster than μ_k^{-1} does. Especially, (3.4) allows one to choose a sequence $\boldsymbol{s}$ that is simply a positive constant or one that tends to infinity.

Using (3.4) we now have $\boldsymbol{b}^\star \in \ell_{\boldsymbol{\mu}}^1 \hookrightarrow \ell_{\boldsymbol{\mu}}^\infty \hookrightarrow \ell_{\boldsymbol{s}^{-1}}^\infty$ and since $\mathcal{F}'(\boldsymbol{b}^\star) \in \ell^\infty$, we also have $\boldsymbol{s} \cdot \mathcal{F}'(\boldsymbol{b}^\star) \in \ell_{\boldsymbol{s}^{-1}}^\infty$. Hence, the term appearing as the argument in the soft-thresholding operator lies in $\ell_{\boldsymbol{s}^{-1}}^\infty$. Now, for $\boldsymbol{v} \in \ell_{\boldsymbol{s}^{-1}}^\infty$, we have that

$$\|\boldsymbol{v}\|_{\ell_{\boldsymbol{s}^{-1}}^\infty} = \max_{k \in \Lambda} s_k^{-1} |v_k| < \infty$$

while the elements of $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})$ are given by

$$\operatorname{sgn}(v_k) \max\bigl\{0, |v_k| - s_k w_k\bigr\}.$$

Since we know that the terms $s_k^{-1}|v_k|$ are bounded while the w_k tend to infinity, it follows that the sequence $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})$ has *finite support* and we thus also have $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}) \in \ell_{\boldsymbol{\mu}}^1$. In view of (3.3), we will consider the soft-threshold operator as a map $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}} \colon \ell_{\boldsymbol{s}^{-1}}^\infty \to \ell_{\boldsymbol{\mu}}^1$ from here on out.

Remark 3.6. We note that, if we have $1 < p, q < \infty$ with $1/p + 1/q = 1$ and $\mathcal{F} \colon \ell^p \to \mathbb{R}$, then all the previous results also hold by replacing ℓ^1 with ℓ^p and ℓ^∞ with ℓ^q. In this case, instead of assuming that $\boldsymbol{w}$ tends to infinity, it suffices to assume that 0 is not an accumulation point of $\boldsymbol{w}$. Hence, $\boldsymbol{\mu}$ also needs not tend to infinity but it still must be bounded away from zero uniformly. For $p = 2$, we thus essentially recover the classic setting considered in works such as [2, 8, 13, 22]. Indeed, since $\mathcal{R}(\boldsymbol{b}) = \infty$ for all $\boldsymbol{b} \in \ell^2 \setminus \ell_{\boldsymbol{\mu}}^1$, the minimisation task

$$\text{minimise} \quad \mathcal{F}(\boldsymbol{b}) + \mathcal{R}(\boldsymbol{b}) \quad \text{over} \quad \boldsymbol{b} \in \ell^2$$

with $\mathcal{F} \colon \ell^2 \to \mathbb{R}$ is obviously equivalent to the setting given by

$$\text{minimise} \quad \mathcal{F}(\boldsymbol{b}) + \mathcal{R}(\boldsymbol{b}) \quad \text{over} \quad \boldsymbol{b} \in \ell_{\boldsymbol{\mu}}^1.$$

4. Optimisation Methods.

Using the fixed-point equation (3.3), we now discuss the optimisation methods that we will utilise to solve our problem (2.5). Before we investigate the possibility of second order methods, we introduce versions of two well-known first order methods adapted to our non-reflexive, non-smooth Banach space setting.

4.1. Simple fixed point methods.

First, we can directly use (3.3) to define the fixed-point iteration

$$\boldsymbol{b}_j := \boldsymbol{T}_{\boldsymbol{s}_j\cdot\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - \boldsymbol{s}_j \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big)$$

starting from some initial value $\boldsymbol{b}_0$. With some strategy for choosing the step sizes $\boldsymbol{s}_j \in \mathbb{R}^{\Lambda}_{\geq 0}$ this will yield a kind of ℓ^1 space version of the known ISTA method, cf. [8]. Note that the ℓ^1 space setting means that step size strategies commonly employed in the Hilbert space setting are not necessarily justified. For example, the strategy used in [2, 20], derives from the fact that the iterate $\boldsymbol{b}_j$ defined by

$$\boldsymbol{b}_j := \boldsymbol{T}_{\lambda_j\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - \lambda_j\mathcal{F}'(\boldsymbol{b}_{j-1})\big)$$

with a scalar step size $\lambda_j \in \mathbb{R}_{>0}$ indeed is the minimiser of the surrogate functional

$$J_j(\boldsymbol{b}) := \mathcal{F}(\boldsymbol{b}_{j-1}) + \big\langle \mathcal{F}'(\boldsymbol{b}_{j-1}), \boldsymbol{b} - \boldsymbol{b}_{j-1}\big\rangle + \mathcal{R}(\boldsymbol{b}) + \frac{1}{2\lambda_j}\|\boldsymbol{b} - \boldsymbol{b}_{j-1}\|^2_{\ell^2}.$$

Now, if $\mathcal{F}'$ is Lipschitz with respect to the ℓ^2-norm, this surrogate provably fulfils $J_j(\boldsymbol{b}) \geq J(\boldsymbol{b})$ when λ_j is chosen small enough. However, in our setting we generally only might have that $\mathcal{F}'$ is Lipschitz with respect to the stronger ℓ^1-norm and hence cannot guarantee that $J_j(\boldsymbol{b}) \geq J(\boldsymbol{b})$ holds even if λ_j is chosen arbitrarily close to 0.

An obvious strategy for choosing the step size is to choose a fixed base step size $\boldsymbol{s} \in \mathbb{R}^{\Lambda}_{\geq 0}$ that fulfils (3.4) and then scale it in each step with some step size multiplier $\lambda_j \in \mathbb{R}_{>0}$, i.e. one uses

$$\boldsymbol{s}_j := \lambda_j\boldsymbol{s}.$$

A simple heuristic approach for determining a suitable step size multiplier is given in the following Algorithm 4.1. In it, to determine the step size multiplier for every iterate, one first tries a step using the initial or previous step size multiplier, if taking this step does not reduce the value of the functional, one successively halves the multiplier until it does (lines 5–8). Then, if one did not need to decrease the multiplier, one doubles the multiplier if this manages to decrease the value of the functional sufficiently more (lines 10–15). Finally, if one did not increase the multiplier, one halves it if this still manages to decrease the value of the functional sufficiently much (lines 16–20). The parameter, which controls when a decrease is sufficient, is the greediness parameter $\sigma \in [0, 1]$. For σ close to zero it only allows the step size multiplier to double, if this also nearly doubles the decrease, while for σ close to one it also allows the step size multiplier to double, as long as the decrease stays the same.

Next, by applying the acceleration from [24] to this version of the ISTA method, we arrive at the following non-Hilbert space version of the FISTA method, cf. [2], given in Algorithm 4.2. Note that compared to the ISTA method, the FISTA method as stated will not guarantee strict monotonicity of the values of the functional $J(\boldsymbol{b}_j)$. To ensure strict monotonicity, we can modify it to reject any step where monotonicity would be violated and restart the acceleration. Since the first iterate computed after (re)starting the acceleration is precisely a normal ISTA iterate this indeed quarantees strict monotonicity and shows that at worst, when such a restarted FISTA method is restarting for every iterate, it coincides with the ISTA method.

Algorithm 4.1 ISTA: Iterated Soft-Thresholding Algorithm

```
Require: Initial value $\boldsymbol{b}_0$, initial step size multiplier $\lambda_0 > 0$, base step size $\boldsymbol{s}$,
         greediness parameter $\sigma \in [0,1]$ for determining step size multiplier
 1: for $j \leftarrow 1, 2, 3, \ldots$ do
 2:     $\lambda_j \leftarrow \lambda_{j-1}$
 3:     $\boldsymbol{b}_j \leftarrow \boldsymbol{T}_{\lambda_j \boldsymbol{s}\cdot\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - \lambda_j \boldsymbol{s} \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big)$
 4:     grow $\leftarrow$ true
 5:     while $J(\boldsymbol{b}_j) \geq J(\boldsymbol{b}_{j-1})$ do
 6:         $\lambda_j \leftarrow \frac{1}{2}\lambda_j$
 7:         $\boldsymbol{b}_j \leftarrow \boldsymbol{T}_{\lambda_j \boldsymbol{s}\cdot\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - \lambda_j \boldsymbol{s} \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big)$
 8:         grow $\leftarrow$ false
 9:     shrink $\leftarrow$ true
10:     if grow = true then
11:         $\boldsymbol{c} \leftarrow \boldsymbol{T}_{2\lambda_j \boldsymbol{s}\cdot\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - 2\lambda_j \boldsymbol{s} \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big)$
12:         if $J(\boldsymbol{c}) \leq J(\boldsymbol{b}_{j-1}) + \frac{2}{\sigma+1}\big(J(\boldsymbol{b}_j) - J(\boldsymbol{b}_{j-1})\big)$ then
13:             $\lambda_j \leftarrow 2\lambda_j$
14:             $\boldsymbol{b}_j \leftarrow \boldsymbol{c}$
15:             shrink $\leftarrow$ false
16:     if shrink = true then
17:         $\boldsymbol{c} \leftarrow \boldsymbol{T}_{\frac{1}{2}\lambda_j \boldsymbol{s}\cdot\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - \frac{1}{2}\lambda_j \boldsymbol{s} \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big)$
18:         if $J(\boldsymbol{c}) < J(\boldsymbol{b}_{j-1}) + \frac{\sigma+1}{2}\big(J(\boldsymbol{b}_j) - J(\boldsymbol{b}_{j-1})\big)$ then
19:             $\lambda_j \leftarrow \frac{1}{2}\lambda_j$
20:             $\boldsymbol{b}_j \leftarrow \boldsymbol{c}$
```

Algorithm 4.2 FISTA: Fast Iterated Soft-Thresholding Algorithm

```
Require: Initial value $\boldsymbol{b}_0$, initial step size multiplier $\lambda_0 > 0$, base step size $\boldsymbol{s}$,
         greediness parameter $\sigma \in [0,1]$ for determining step size multiplier
$t_0 \leftarrow 1$
$\boldsymbol{q}_0 \leftarrow \boldsymbol{b}_0$
for $j \leftarrow 1, 2, 3, \ldots$ do
    Compute lines 2–20 in Algorithm 4.1 with $\boldsymbol{b}_{j-1}$ replaced by $\boldsymbol{q}_{j-1}$
    $t_j \leftarrow \frac{1}{2}\big(1 + (1 + 4t_{j-1}^2)^{1/2}\big)$
    $\boldsymbol{q}_j \leftarrow \boldsymbol{b}_j + \frac{t_{j-1}-1}{t_j}(\boldsymbol{b}_j - \boldsymbol{b}_{j-1})$
```

4.2. Newton differentiability of the soft-threshold operator. As is well known and as we will see later on in the numerical experiments, the simple fixed-point methods given above are fairly slow in their convergence. However, as was done in e.g. [13, 22], we wish to consider semismooth Newton methods for solving (3.3), cf. [5]. To justify this, we extend the result from [13, Proposition 3.3] asserting that the soft-threshold operator is Newton differentiable as a map from ℓ^p to ℓ^r for abitrary $1 \leq p < \infty$ and $1 \leq r \leq \infty$ by proving that the soft-threshold operator is indeed also Newton differentiable as a map from $\ell^\infty_{\boldsymbol{s}^{-1}}$ to $\ell^1_{\boldsymbol{\mu}}$.

Lemma 4.1. *The soft-threshold operator* $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}} \colon \ell^\infty_{\boldsymbol{s}^{-1}} \to \ell^1_{\boldsymbol{\mu}}$ *is Newton differentiable and* $t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}} \colon \ell^\infty_{\boldsymbol{s}^{-1}} \to \mathcal{L}(\ell^\infty_{\boldsymbol{s}^{-1}}, \ell^1_{\boldsymbol{\mu}})$ *defined by*

$$t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})[\boldsymbol{h}] := \left(\begin{cases} h_k, & \textit{when } |v_k| > s_k w_k, \\ 0, & \textit{when } |v_k| \leq s_k w_k, \end{cases} \right)_{k\in\Lambda}$$

is a slanting function for $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}$ *on the whole of* $\ell^\infty_{\boldsymbol{s}^{-1}}$.

Proof. As the elements of $\boldsymbol{w}$ tend to infinity, when Λ is not finite, we know that

$$\Lambda_{\boldsymbol{v}} := \left\{ k \in \Lambda : w_k < \|\boldsymbol{v}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} + 1 \right\}$$

is a finite set. For any $\boldsymbol{h} \in \ell^\infty_{\boldsymbol{s}^{-1}}$ with $\|\boldsymbol{h}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} \leq 1$ and any $k \in \Lambda \setminus \Lambda_{\boldsymbol{v}}$, we then have that

$$s_k^{-1}|v_k + h_k| \leq \|\boldsymbol{v} + \boldsymbol{h}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} \leq \|\boldsymbol{v}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} + 1 \leq w_k$$

and, therefore, the elements of $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})$, $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})$ and $t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})[\boldsymbol{h}]$ at index k are all zero.

Now, we split the set $\Lambda_{\boldsymbol{v}}$ into the active, edge-case and inactive indices:

$$\begin{aligned} \Lambda^{\mathrm{a}}_{\boldsymbol{v}} &:= \left\{ k \in \Lambda_{\boldsymbol{v}} : |v_k| > s_k w_k \right\}, \\ \Lambda^{\mathrm{e}}_{\boldsymbol{v}} &:= \left\{ k \in \Lambda_{\boldsymbol{v}} : |v_k| = s_k w_k \right\}, \\ \Lambda^{\mathrm{i}}_{\boldsymbol{v}} &:= \left\{ k \in \Lambda_{\boldsymbol{v}} : |v_k| < s_k w_k \right\}. \end{aligned}$$

Obviously, for any $\boldsymbol{h} \in \ell^\infty_{\boldsymbol{s}^{-1}}$ the difference of the elements of $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})$ and $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})$ at index k is equal to $t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})[\boldsymbol{h}]$ at index k for $k \in \Lambda^{\mathrm{e}}_{\boldsymbol{v}}$. Next, we introduce

$$\delta := \min_{k\in\Lambda^{\mathrm{a}}_{\boldsymbol{v}}\cup\Lambda^{\mathrm{i}}_{\boldsymbol{v}}} \left| |v_k| - s_k w_k \right| > 0.$$

For any $\boldsymbol{h} \in \ell^\infty_{\boldsymbol{s}^{-1}}$ with $\|\boldsymbol{h}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} \leq \delta$ we have that the elements of $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})$, $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})$ and $t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})[\boldsymbol{h}]$ at index k are all zero for $k \in \Lambda^{\mathrm{i}}_{\boldsymbol{v}}$. Similarly, the difference of the elements of $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})$ and $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v})$ at index k is equal to $t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})[\boldsymbol{h}]$ at index k for $k \in \Lambda^{\mathrm{a}}_{\boldsymbol{v}}$.

Combining all this shows that we have

$$\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h}) - \boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}) - t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})[\boldsymbol{h}] = \boldsymbol{0} \in \ell^1_\mu$$

for any $\boldsymbol{h} \in \ell^\infty_{\boldsymbol{s}^{-1}}$ with $\|\boldsymbol{h}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} \leq \min\{1, \delta\}$ and thus

$$\lim_{\|\boldsymbol{h}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}} \to 0} \frac{\left\| \boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h}) - \boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}) - t^\circ_{\boldsymbol{s}\cdot\boldsymbol{w}}(\boldsymbol{v}+\boldsymbol{h})[\boldsymbol{h}] \right\|_{\ell^1_\mu}}{\|\boldsymbol{h}\|_{\ell^\infty_{\boldsymbol{s}^{-1}}}} = 0$$

holds, proving that $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}$ is Newton differentiable and $t^{\circ}_{\boldsymbol{s}\cdot\boldsymbol{w}}$ is a slanting function for $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}$ on the whole of $\ell^{\infty}_{\boldsymbol{s}^{-1}}$. ■

4.3. Semismooth methods. Since $\mathcal{F}\colon \ell^1 \to \mathbb{R}$ is twice Fréchet differentiable with $\mathcal{F}''$ being locally Lipschitz, cf. Remark 3.2, it is possible to use the Newton differentiabilty of $\boldsymbol{T}_{\boldsymbol{s}\cdot\boldsymbol{w}}$ and a chain rule for Newton differentiability to derive a semismooth Newton method for solving (3.3), see [22].

For this, we introduce the indicator sequences as follows: Given some iterate $\boldsymbol{b}_{j-1}$ and step length $\boldsymbol{s}_j$, we define the upper active indicator by

$$\boldsymbol{i}_j^{\mathrm{a}+} := \left(\begin{cases} 1, & \text{when } \big[\boldsymbol{b}_{j-1} - \boldsymbol{s}_j \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big]_k > [\boldsymbol{s}_j \cdot \boldsymbol{w}]_k, \\ 0, & \text{else,} \end{cases}\right)_{k\in\Lambda}$$

and the lower active indicator by

$$\boldsymbol{i}_j^{\mathrm{a}-} := \left(\begin{cases} 1, & \text{when } \big[\boldsymbol{b}_{j-1} - \boldsymbol{s}_j \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big]_k < -[\boldsymbol{s}_j \cdot \boldsymbol{w}]_k, \\ 0, & \text{else,} \end{cases}\right)_{k\in\Lambda}.$$

The active and inactive indicators are now defined by $\boldsymbol{i}_j^{\mathrm{a}} := \boldsymbol{i}_j^{\mathrm{a}+} + \boldsymbol{i}_j^{\mathrm{a}-}$ and $\boldsymbol{i}_j^{\mathrm{i}} := \boldsymbol{i} - \boldsymbol{i}_j^{\mathrm{a}}$, where $\boldsymbol{i} := (1)_{k\in\Lambda}$. The corresponding active sets are obviously given by

$$\Lambda_j^t := \Big\{k \in \Lambda : \big[\boldsymbol{i}_j^t\big]_k = 1\Big\}$$

for $t \in \{\mathrm{a}+, \mathrm{a}-, \mathrm{a}, \mathrm{i}\}$.

Now, given the iterate $\boldsymbol{b}_{j-1}$ and step length $\boldsymbol{s}_j$, the next iterate $\boldsymbol{b}_j$ of the semismooth Newton method applied to the equation

$$\boldsymbol{0} = \boldsymbol{b} - \boldsymbol{T}_{\boldsymbol{s}_j\cdot\boldsymbol{w}}\big(\boldsymbol{b} - \boldsymbol{s}_j \cdot \mathcal{F}'(\boldsymbol{b})\big)$$

is defined by

$$\boldsymbol{b}_j := \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{b}_{j-1} - \boldsymbol{d}_j,$$

where $\boldsymbol{d}_j$ fulfils the equations

$$\begin{aligned} \boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{d}_j &= \boldsymbol{0}, \\ \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}''(\boldsymbol{b}_{j-1})\big[\boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{d}_j\big] &= \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}'(\boldsymbol{b}_{j-1}) \pm \boldsymbol{i}_j^{\mathrm{a}\pm} \cdot \boldsymbol{w} - \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}''(\boldsymbol{b}_{j-1})\big[\boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{b}_{j-1}\big], \end{aligned} \tag{4.1}$$

where $\pm\boldsymbol{i}_j^{\mathrm{a}\pm} \cdot \boldsymbol{w} = \boldsymbol{i}_j^{\mathrm{a}+} \cdot \boldsymbol{w} - \boldsymbol{i}_j^{\mathrm{a}-} \cdot \boldsymbol{w}$, cf. [22]. Note that the first equation in (4.1) directly prescribes the value of 0 to $\boldsymbol{d}_j$ at all indices in the inactive set Λ_j^{i}, while the second equation in (4.1) only depends on the values of $\boldsymbol{d}_j$ at indices in the active set Λ_j^{a}.

Moreover, since the active set is of finite cardinality, the linear map

$$\boldsymbol{v} \mapsto \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}''(\boldsymbol{b}_{j-1})\big[\boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{v}\big]$$

can be understood as a square matrix $\boldsymbol{H}_j$ which maps the finite dimensional space $\mathbb{R}^{\Lambda_j^{\mathrm{a}}}$ into itself. Therefore, as the right-hand side of the second equation in (4.1) also lies in $\mathbb{R}^{\Lambda_j^{\mathrm{a}}}$, we know that $\boldsymbol{d}_j$ is uniquely defined if and only if $\boldsymbol{H}_j$ has full rank. However, while Lemma 3.1 shows that, as is well known for such problems, the derivative $\mathcal{F}'(\boldsymbol{b})$ may be computed efficiently by simply solving two boundary value problems, the computation of the Hessian $\mathcal{F}''(\boldsymbol{b})$ is known to be more expensive. In [22], the authors thus propose to replace the exact Hessian with an approximation, i.e. the use of a semismooth quasi-Newton method.

In contrast to this, given the structure of our problem, we consider a Gauss-Newton type of modification instead. That is, given the iterate $\boldsymbol{b}_{j-1}$, one defines the local approximation of $\mathcal{F}$ by

$$\mathcal{F}_j(\boldsymbol{b}) := \frac{1}{2}\big\|M(\boldsymbol{b}_{j-1}) + M'(\boldsymbol{b}_{j-1})[\boldsymbol{b} - \boldsymbol{b}_{j-1}]\big\|^2_{L^2(\Omega)},$$

compare (2.3) for the precise definition of the operators M and $\mathcal{F}$. Then using the step length $\boldsymbol{s}_j$, one computes the next iterate $\boldsymbol{b}_j$ by a single step of the semismooth Newton method applied to the equation

$$\boldsymbol{0} = \boldsymbol{b} - \boldsymbol{T}_{\boldsymbol{s}_j\cdot\boldsymbol{w}}\big(\boldsymbol{b} - \boldsymbol{s}_j \cdot \mathcal{F}_j'(\boldsymbol{b})\big)$$

from the iterate $\boldsymbol{b}_{j-1}$. Since by construction $\mathcal{F}_j(\boldsymbol{b}_{j-1}) = \mathcal{F}(\boldsymbol{b}_{j-1})$ and $\mathcal{F}_j'(\boldsymbol{b}_{j-1}) = \mathcal{F}'(\boldsymbol{b}_{j-1})$ hold, the active and inactive indicators and sets are the same as before and we arrive at

$$\boldsymbol{b}_j := \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{b}_{j-1} - \boldsymbol{d}_j,$$

where $\boldsymbol{d}_j$ instead fulfils the equations

$$\begin{aligned} \boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{d}_j &= \boldsymbol{0}, \\ \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j''(\boldsymbol{b}_{j-1})\big[\boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{d}_j\big] &= \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j'(\boldsymbol{b}_{j-1}) \pm \boldsymbol{i}_j^{\mathrm{a}\pm} \cdot \boldsymbol{w} - \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j''(\boldsymbol{b}_{j-1})\big[\boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{b}_{j-1}\big]. \end{aligned} \tag{4.2}$$

As $\mathcal{F}_j$ is a quadratic polynomial over the Banach space ℓ^1, its second order Fréchet derivative is simply given by

$$\big\langle \mathcal{F}_j''(\boldsymbol{b}_{j-1})[\boldsymbol{v}_1], \boldsymbol{v}_2\big\rangle = \mathcal{F}_j''(\boldsymbol{b}_{j-1})[\boldsymbol{v}_1, \boldsymbol{v}_2] = \big(M'(\boldsymbol{b}_{j-1})[\boldsymbol{v}_2], M'(\boldsymbol{b}_{j-1})[\boldsymbol{v}_1]\big)_{L^2(\Omega)}.$$

Hence, the second order Fréchet derivative results in a symmetric and positive semidefinite matrix $\boldsymbol{H}_j$ when one restricts it onto the finite subspace $\mathbb{R}^{\Lambda_j^{\mathrm{a}}}$.

However, this Gauss-Newton type approach still leaves us with the challenge of solving a linear system of equations with a symmetric and positive semidefinite matrix. To overcome this, it is natural to consider a Levenberg-Marquardt type stabilisation of the system matrix. As a suitable step size $\boldsymbol{s}_j$ is needed to be determine the active sets, we propose that the stabilisation is derived by considering the fixed-point iterate in the ISTA algorithm using the same step size. For this, we notice that the fixed-point step with step size $\boldsymbol{s}_j$,

$$\boldsymbol{b}_j := \boldsymbol{T}_{\boldsymbol{s}_j\cdot\boldsymbol{w}}\big(\boldsymbol{b}_{j-1} - \boldsymbol{s}_j \cdot \mathcal{F}'(\boldsymbol{b}_{j-1})\big),$$

also can be given by

$$\boldsymbol{b}_j := \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{b}_{j-1} - \boldsymbol{d}_j,$$

where $\boldsymbol{d}_j$ instead fulfils the equations

$$\begin{aligned} \boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{d}_j &= \boldsymbol{0}, \\ \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{s}_j^{-1} \cdot \boldsymbol{d}_j &= \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j'(\boldsymbol{b}_{j-1}) \pm \boldsymbol{i}_j^{\mathrm{a}\pm} \cdot \boldsymbol{w}. \end{aligned} \tag{4.3}$$

Specifically, we propose that one blends the equation (4.2), defining a Gauss-Newton type update, in a sigmoidal manner with the equation (4.3), defining the fixed-point update. For $\kappa \in \mathbb{R}$, we may combine the equations using the weights $\frac{1}{1+2^\kappa}$ and $\frac{1}{2^{-\kappa}+1}$ yielding the equations

$$\begin{aligned} &\boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{d}_j = \boldsymbol{0}, \\ &\frac{1}{1+2^\kappa} \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j''(\boldsymbol{b}_{j-1}) \big[\boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{d}_j\big] + \frac{1}{2^{-\kappa}+1} \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{s}_j^{-1} \cdot \boldsymbol{d}_j \\ &\qquad = \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j'(\boldsymbol{b}_{j-1}) \pm \boldsymbol{i}_j^{\mathrm{a}\pm} \cdot \boldsymbol{w} - \frac{1}{1+2^\kappa} \boldsymbol{i}_j^{\mathrm{a}} \cdot \mathcal{F}_j''(\boldsymbol{b}_{j-1}) \big[\boldsymbol{i}_j^{\mathrm{i}} \cdot \boldsymbol{b}_{j-1}\big]. \end{aligned} \tag{4.4}$$

for computing a Levenberg-Marquardt type update. This symmetric and positive definite equation simply can be solved approximately by the CG-method for example. By using a simple strategy for de- and increasing the stabilisation parameter κ, we arrive at the method described in Algorithm 4.3, called the Active Set Iterated Soft-Threshold Algorithm (ASISTA).

Algorithm 4.3 ASISTA: Active Set Iterated Soft-Threshold Algorithm

```
Require: Initial value $\boldsymbol{b}_0$, initial step size multiplier $\lambda_0 > 0$, base step size $\boldsymbol{s}$,
         greediness parameter $\sigma \in [0, 1]$ for determining step size multiplier
$\kappa \leftarrow 0$
$\boldsymbol{d}_0 \leftarrow \boldsymbol{0}$
for $j \leftarrow 1, 2, 3, \dots$ do
    Compute lines 2–20 in Algorithm 4.1
    Compute the indicators $\boldsymbol{i}_j^{\mathrm{a}+}$, $\boldsymbol{i}_j^{\mathrm{a}-}$, $\boldsymbol{i}_j^{\mathrm{a}}$ and $\boldsymbol{i}_j^{\mathrm{i}}$
    stepok $\leftarrow$ false
    shrink $\leftarrow$ true
    while stepok = false do
        $\boldsymbol{d}_j \leftarrow \frac{1}{1+2^\kappa} \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{d}_{j-1} + \frac{1}{2^{-\kappa}+1} (\boldsymbol{b}_{j-1} - \boldsymbol{b}_j)$
        Update $\boldsymbol{d}_j$ to approximately fulfil (4.4) using CG
        $\boldsymbol{c}_j \leftarrow \boldsymbol{i}_j^{\mathrm{a}} \cdot \boldsymbol{b}_{j-1} - \boldsymbol{d}_j$
        if $J(\boldsymbol{b}_{j-1}) \leq J(\boldsymbol{c}_j)$ then
            $\kappa \leftarrow \kappa + 1$
            shrink $\leftarrow$ false
        else
            if $J(\boldsymbol{b}_j) < J(\boldsymbol{c}_j)$ or CG did not converge sufficiently then
                shrink $\leftarrow$ false
            stepok $\leftarrow$ true
    $\boldsymbol{b}_j \leftarrow \boldsymbol{c}_j$
    if shrink = true then
        $\kappa \leftarrow \kappa - 1$
```

Again, by simply applying the acceleration from [24] to ASISTA, we also introduce the corresponding method given in Algorithm 4.4, which we call the Fast Active Set Iterated Soft-Threshold Algorithm (FASISTA). As is with FISTA, FASISTA also does not ensure strict monotonicity as stated but can be modified to do so by rejecting any step where monotonicity would be violated and restarting the acceleration.

Algorithm 4.4 FASISTA: Fast Active Set Iterated Soft-Threshold Algorithm

```
Require: Initial value $\boldsymbol{b}_0$, initial step size multiplier $\lambda_0 > 0$, base step size $\boldsymbol{s}$,
         greediness parameter $\sigma \in [0,1]$ for determining step size multiplier
$\kappa \leftarrow 0$
$\boldsymbol{d}_0 \leftarrow \boldsymbol{0}$
for $j \leftarrow 1, 2, 3, \dots$ do
    Compute lines 4–21 in Algorithm 4.3 with $\boldsymbol{b}_{j-1}$ replaced by $\boldsymbol{q}_{j-1}$
    $t_j \leftarrow \frac{1}{2}\big(1 + (1 + 4t_{j-1}^2)^{1/2}\big)$
    $\boldsymbol{q}_j \leftarrow \boldsymbol{b}_j + \frac{t_{j-1}-1}{t_j}(\boldsymbol{b}_j - \boldsymbol{b}_{j-1})$
```

Remark 4.2. Again, we note that all the results in this section also hold when one has $\mathcal{F}\colon \ell^p \to \mathbb{R}$ with $1 < p, q < \infty$ that fulfils $1/p + 1/q = 1$, if one replaces ℓ^1 with ℓ^p and ℓ^∞ with ℓ^q, cf. Remark 3.6. For the case $p = 2$, we then can observe that the methods defined by the equations (4.1), (4.2) and (4.4) can also be derived as inexact proximal Newton-type methods, see [19]. For this, one simply approximates the smooth part of the functional $\mathcal{F}$ in each step as is done in the Newton, Gauss-Newton or Levenberg-Marquardt methods and then approximately solves the subproblem using a single step of the semismooth Newton method from [13]. Therefore in this case, ASISTA might be considered to be an inexact proximal Levenberg-Marquardt-type method.

5. Remarks on discretisation.

To solve the optimal control problem (2.5) with the optimisation algorithms, we need to discretise the partial differential equations as well as choose an appropriate bounded sequence $\boldsymbol{\psi}$ for the expansion.

For the sake of simplicity, we assume from here on that the domain Ω is the unit square $\Omega := (0,1)^2$. Then, for some given $N \in \mathbb{N}$, we denote the set of all square elements that are formed by subdividing the square into N^2 square elements of side length $h := N^{-1}$ by $\mathcal{Q}_N$.

5.1. Discretisation of the parameter-to-state mapping.

We straightforwardly utilise bilinear finite elements and discretise the partial differential equation in their weak form using the Galerkin method. For this, we introduce the space of bilinear finite element functions

$$V_N := \big\{u \in C(\overline{\Omega}) : u \text{ is bilinear on every element } Q \in \mathcal{Q}_N \text{ and } u = 0 \text{ on } \Gamma\big\}$$

and let $(\phi_i)_{i=1}^n$ with $n = (N-1)^2$ be the nodal basis of V_N. For the discretisation of the diffusion coefficient a, we also introduce the space of element-wise constant finite element functions

$$W_N := \big\{a \in L^\infty(\Omega) : a \text{ is constant on every element } Q \in \mathcal{Q}_N\big\}$$

and let $(\chi_i)_{i=1}^m$ with $m = N^2$ be the basis of W_N that is made up of all the indicator functions of the elements, i.e. the functions $\mathbf{1}_Q$ for $Q \in \mathcal{Q}_N$. Now, for functions $v \in V_N$ and $w \in W_N$, we denote their coefficients with respect to the correspondig bases by $\widehat{v}$ and $\widehat{w}$.

We will assume that the sequence defining the expansion lies in W_N. That is the expansion $a = E(\boldsymbol{b})$ can be defined by $\widehat{a} = \boldsymbol{E}\boldsymbol{b}$ for all $\boldsymbol{b} \in \ell^1$, where $\boldsymbol{E} \in \mathbb{R}^{m \times \Lambda}$ is a (possibly semi-infinite) matrix. Moreover, for a more concise exposition, we will assume that $u_g - u_d$ lies in V_N, where u_g is the H^1-extension of g and u_d the measurement. In practise this might be enforced by replacing u_g and u_d with some approximations of them in V_N.

Considering the weak formulation of (2.1), we define the stiffness matrix with coefficient $a \in W_N$ by

$$\boldsymbol{A}_a = \left[\int_\Omega a(\boldsymbol{x}) \big\langle \nabla \phi_j(\boldsymbol{x}), \nabla \phi_i(\boldsymbol{x}) \big\rangle \,\mathrm{d}\boldsymbol{x} \right]_{i,j} \in \mathbb{R}^{n \times n}$$

and the right-hand side using the H^1-extension $u_g \in H^1(\Omega)$ of g by

$$\boldsymbol{f}_a = \left[\int_\Omega f(\boldsymbol{x}) \phi_i(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} - \int_\Omega a(\boldsymbol{x}) \big\langle \nabla u_g(\boldsymbol{x}), \nabla \phi_i(\boldsymbol{x}) \big\rangle \,\mathrm{d}\boldsymbol{x} \right]_i \in \mathbb{R}^n.$$

Now, for $\widehat{a} = \boldsymbol{E}\boldsymbol{b}$, we have that the Galerkin approximation of (2.1) is given by

$$u = u_0 + u_g, \quad \text{where} \quad \widehat{u}_0 = \boldsymbol{A}_a^{-1} \boldsymbol{f}_a.$$

Next, we introduce the mass matrix

$$\boldsymbol{M} = \left[\int_\Omega \phi_j(\boldsymbol{x}) \phi_i(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \right]_{i,j} \in \mathbb{R}^{n \times n},$$

with which we can compute the discretised data fidelity (2.3) as

$$\mathcal{F}(\boldsymbol{b}) = \frac{1}{2} (\widehat{u_0} + \widehat{u}_g - \widehat{u}_d)^\mathsf{T} \boldsymbol{M} (\widehat{u}_0 + \widehat{u}_g - \widehat{u}_d),$$

and the Galerkin approximation of the adjoint state p by

$$\widehat{p} = \boldsymbol{A}_a^{-1} \boldsymbol{M} (\widehat{u_0} + \widehat{u}_g - \widehat{u}_d).$$

Finally, we define the matrix

$$\boldsymbol{W}_{a,u_0} = \left[\int_\Omega a(\boldsymbol{x}) \chi_j(\boldsymbol{x}) \big\langle \nabla (u_0 + u_g)(\boldsymbol{x}), \nabla \phi_i(\boldsymbol{x}) \big\rangle \,\mathrm{d}\boldsymbol{x} \right]_{i,j} \in \mathbb{R}^{n \times m}$$

with coefficient $a \in W_N$ and solution $u_0 + u_g$ with $u_0 \in V_N$. Then, it is easy to see that the discretised derivative of the data fidelity as an element of ℓ^∞ is given by

$$\mathcal{F}'(\boldsymbol{b}) = -\boldsymbol{E}^\mathsf{T} \boldsymbol{W}_{a,u_0}^\mathsf{T} \widehat{p}$$

and the discretised second derivative of the approximated data fidelity for $\boldsymbol{b} = \boldsymbol{b}_{j-1}$ is given by

$$\mathcal{F}_j''(\boldsymbol{b})[\boldsymbol{v}_1, \boldsymbol{v}_2] = \boldsymbol{v}_2^\top \boldsymbol{E}^\top \boldsymbol{W}_{a,u_0}^\top \boldsymbol{A}_a^{-\top} \boldsymbol{M} \boldsymbol{A}_a^{-1} \boldsymbol{W}_{a,u_0} \boldsymbol{E} \boldsymbol{v}_1.$$

5.2. Choice of the parameter expansion. In agreement with the preeceding subsection, one has to choose a bounded sequence $\boldsymbol{\psi} = (\psi_k)_{k\in\Lambda} \subset W_N \subset L^\infty(\Omega)$ which then defines the expansion (2.2). In this case, the kth column in the (possibly semi-infinite) matrix $\boldsymbol{E} \in \mathbb{R}^{m\times\Lambda}$ is precisely the coefficients $\widehat{\psi}_k$. Obviously, while there are a myriad of possible expansions, the main point to consider here is that $\log(a)$ is supposed to be approximated by a fairly sparse expansion for all likely diffusion parameter functions $a \in A_{\mathrm{ad}}$.

However, there is at another point which should be taken into account. As can be seen in the preceeding subsection, both the expansion $\boldsymbol{E}$ as well as its transpose $\boldsymbol{E}^\top$ will need to be applied during every iteration of the optimisation. This means that expansions whose applications have a computational complexity which scales nearly linearly in $\max\{|\Lambda|, m\}$ are preferential to those that scale like the product $|\Lambda|m$. This motivates the utilisation of expansions such as wavelet and wavelet-like expansions or Fourier-type series, when the logarithm of the diffusion parameter is likely to be cartoon-like or very smooth, respectively. For our numerical experiments, we will consider the following two choices:

- We choose to only consider the simplest wavelet expansion, that is the isotropic two-dimensional Haar wavelets. For sake of simplicity, we restrict the possible N to a power of two, i.e. $N = 2^L$. Note that the Haar wavelets are scaled to have a L^∞-norm of 1, so that they form a bounded sequence. Moreover, the application of both $\boldsymbol{E}$ as well as $\boldsymbol{E}^\top$ have a log-linear computational complexity in $m = N^2 = 4^L$. For this expansion, we choose to define $\boldsymbol{\mu}$ by setting $\mu_k = 2^{\ell-1}$ for all $1 \le k \le N^2$, for which ψ_k is a wavelet on level ℓ.

  Remark 5.1. In general, the use of a Haar wavelet expansion is not necessarily optimal. If the logarithm of the diffusion parameter is a cartoon-like function, then curvelets, contourlets and similar bases and frames are likely to have sparser expansions, see [4, 9, 14]. We also want to point out that one can consider a general domain by constructing wavelets over any type of finite element discretisation using the approach of Tausch and White, see [25].

- We consider a two-dimensional discrete cosine series expansion as a simple example for a Fourier-type expansion. To this end, let $k = r_k + N(s_k - 1)$ for all $1 \le k \le N^2$ with $1 \le r_k, s_k \le N$, then we define ψ_k by

  $$\psi_k = \sum_{Q\in\mathcal{Q}_N} \cos\big(\pi(r_k - 1)c_{Q,1}\big) \cos\big(\pi(s_k - 1)c_{Q,2}\big)\mathbf{1}_Q,$$

  where $(c_{Q,1}, c_{Q,2})$ denotes the coordinates of the centre and $\mathbf{1}_Q$ the indicator function of an element $Q \in \mathcal{Q}_N$. Specifically, the expansion that this finite sequence yields is a rescaled two-dimensional version of the transform known as the type III discrete cosine transform (DCT-III) or inverse of the type II discrete cosine transform (DCT-II). Therefore, by rescaling it, one can efficiently compute the application of both $\boldsymbol{E}$

as well as $\boldsymbol{E}^{\mathsf{T}}$ using fast cosine transform (FCT) algorithms, that have a log-linear computational complexity in $m = N^2$. For this expansion, we choose to define $\boldsymbol{\mu}$ by setting $\mu_k = \sqrt{r_k^2 + s_k^2}$ for all $1 \le k \le N^2$.

6. Numerical examples.

To illustrate the behaviour of the minimisation methods as well as that of the regularisation, we consider the reconstruction of the two diffusion parameters shown in Figure 1, from here on also referred to as phantoms.

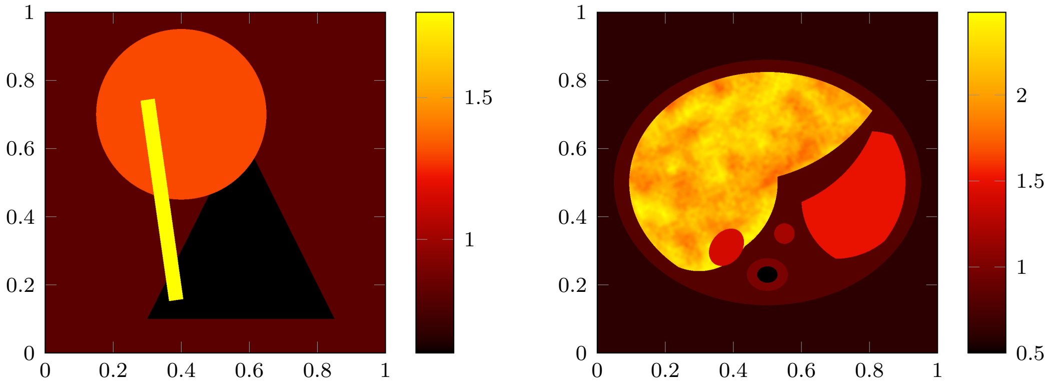

Figure 1. *The two diffusion parameters considered in the numerical examples. The phantom on the left* (geometric phantom) *consists of three superimposed simple geometric shapes, while the phantom on the right is inspired by an abdominal cross-section of a human torso* (torso phantom)*.*

6.1. Comparison of the minimisation methods.

In our first numerical example, we focus on the behaviour of the minimisation methods, and of the step size strategy. We let the right-hand sides of (2.1) be $f = 1$ and $g = 0$ and consider the reconstruction of the geometric phantom. For the minimisation methods, we use N^2 finite elements to represent the state and the coefficient as described in Section 5 with $N = 2^7$.

The synthetic measurement u_d is computed as follows: We compute an approximation u_r of the exact state using N_r^2 bilinear finite elements with $N_r = 2^8 - 1$. This choice ensures that the associated meshes are not nested. The respective solution $u_r \in V_{N_r}$ is then projected into the space V_N by means of the L^2-best approximation, yielding u_c. Then, we define the synthetic measurement by $u_d := u_c + \delta\eta$, where $\eta \in V_N$ indicates white Gaussian noise at the nodes of the elements defining V_N, scaled to fulfil $\|\eta\|_{L^2(\Omega)} = \|u_c\|_{L^2(\Omega)}$. The noise level is set to $\delta := 10^{-3}$. This approach results in a relative L^2-error in the data that approximately equals δ, with any deviation from this stemming from the error made in the L^2-best approximation. In our example, this yields the following relative L^2- and H^1-error in the data,

$$\frac{\|u_d - u_r\|_{L^2(\Omega)}}{\|u_r\|_{L^2(\Omega)}} \approx 0.0010095, \quad \text{and} \quad \frac{\|u_d - u_r\|_{H^1(\Omega)}}{\|u_r\|_{H^1(\Omega)}} \approx 0.0703314.$$

For this first example, we choose the Haar wavelet expansion and consider regularisation weights given by $\boldsymbol{w} := \varrho\boldsymbol{\mu}$. We then use the ISTA, FISTA, ASISTA and FASISTA methods given in Algorithms 4.1, 4.2, 4.3 and 4.4 to minimise the optimal control problem (2.5), where we modify FISTA and FASISTA to restart the acceleration to ensure monotonicity. We apply all four methods for each greediness parameter $\sigma \in \{0.3, 0.4, \dots, 0.9\}$ and each regularisation strength $\varrho \in \{10^{-11}, 10^{-12}, 10^{-13}, 10^{-14}\}$. The initial value for the methods is $\boldsymbol{b}_0 := \mathbf{0}$ and the base step size is $\boldsymbol{s} = \mathbf{1}$. We stop a method when it has solved 42 000 PDEs (forward and adjoint problems). The CG-solver in the ASISTA and FASISTA methods is declared to have converged if the relative residual measured in the ℓ^∞-norm is smaller than 10^{-2} and is otherwise stopped after 50 iterations.

In order to compare the efficiency of the methods, we suggest as the measure of cost to consider how many PDEs (forward and adjoint problems) the method had to solve to arrive at its kth iterate. In Figure 2, we plot the distance between the functional at the kth iterate $J(\boldsymbol{b}_k)$ and the estimated minimum value $J(\boldsymbol{b}^*)$ as a function of total number of PDE solves necessary to compute the kth iterate. The figure shows that the ASISTA and FASISTA methods generally minimise the functional J more effectively, i.e. they needs fewer PDE solves than the ISTA and FISTA methods, with FASISTA generally outperforming ASISTA at leas slightly. Indeed, the figure demonstrates that the ISTA method convergences very slowly and shows that the FISTA method is truly accelerated. Moreover, while the FISTA method comes close to matching the performance of the ASISTA method for $\varrho = 10^{-11}$ and $\varrho = 10^{-12}$ after around 10 000 PDE solves, it is simply outperformed for $\varrho = 10^{-13}$ and $\varrho = 10^{-14}$.

In Figure 2, the reconstruction of the phantom given by the last iterate of the FASISTA method using $\sigma = 0.5$ is also depicted. These reconstructions show that, while the regularisation strengths $\varrho = 10^{-11}$ and $\varrho = 10^{-14}$ are over- and underregularising, $\varrho = 10^{-12}$ and $\varrho = 10^{-13}$ are regularising quite effectively.

Figure 3 shows the distance between the functional at the kth iterate $J(\boldsymbol{b}_k)$ and the estimated minimum value $J(\boldsymbol{b}^*)$ as a function of iteration number k. It is noticeable that ISTA does not seem able to achieve the rate k^{-1} which is known to hold for the classical ℓ^2-setting, but rather a reduced rate $k^{-1/2}$. However, for the strongest regularisation with $\varrho = 10^{-11}$, it seems that FISTA manages to mostly achieve the rate k^{-2} that is known to hold for the classical ℓ^2-setting. On the other hand, FISTA also seems to only achieve the reduced rate k^{-1} for the weakest regularisation with $\varrho = 10^{-14}$, while its behaviour for $\varrho = 10^{-12}$ and $\varrho = 10^{-13}$ lies somewhere in between the two extreme cases. Figure 3 also indicates that the ASISTA and FASISTA methods behave somewhat inversely: For $\varrho = 10^{-14}$ they seem to be able to maintain their steep slope magnitude the longest as k increases, while for $\varrho = 10^{-11}$ they suffer a noticeable decrease in slope magnitude.

6.2. Effectiveness of the regularisation. Our second numerical example focuses on the effectiveness of the regularisation vis-à-vis noise. The setup of this example is the same as for the first example with the following changes: We consider the torso phantom and set $N = 2^9$ and $N_r = 2^{10} - 1$. We consider the six levels of noise $\delta \in \{10^{-1.5}, 10^{-2}, 10^{-2.5}, 10^{-3}, 10^{-3.5}, 10^{-4}\}$, which yield relative L^2- and H^1-errors in the data as shown in Table 1.

For this second example, we consider both the Haar wavelet expansion and the discrete cosine expansion. The regularisation weights in both cases are given by $\boldsymbol{w} := \varrho\boldsymbol{\mu}$ where we choose the regularisation strengths $\varrho \in \{10^{-11}, 10^{-12.5}, 10^{-14}, 10^{-15.5}\}$ for the Haar wavelet expansion and $\varrho \in \{10^{-10.5}, 10^{-12}, 10^{-13.5}, 10^{-15}\}$ for the discrete cosine expansion. We use the FASISTA method with greediness parameter $\sigma = 0.5$ to minimise the optimal control problem (2.5) and stop it after it has solved 5 000 PDEs (forward and adjoint problems).

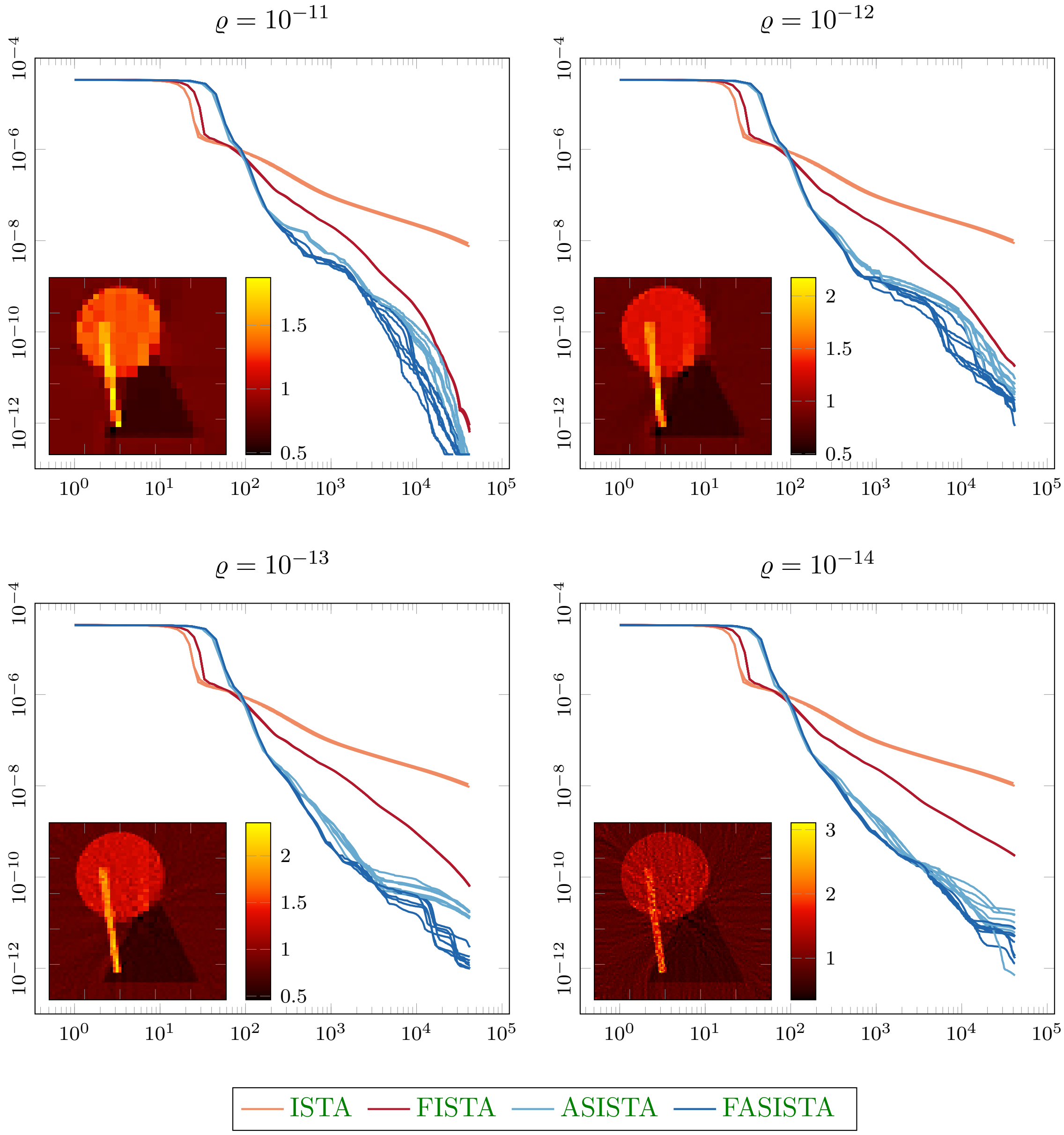

Figure 2. *First example: Distance to the estimated minimum (vertical axis), i.e.* $J(\boldsymbol{b}_k) - J(\boldsymbol{b}^*)$*, as a function of the total number of PDE solves necessary to compute the* k*th iterate (horizontal axis). The four plots show the four different regularisation strengths,* $\varrho = 10^{-11}, 10^{-12}, 10^{-13}, 10^{-14}$*. For each method, the different lines correspond to the different choices for the greediness parameter,* $\sigma = 0.3, 0.4, \ldots, 0.9$*. The reconstruction depicted in the lower left of each axis is the last iterate of FASISTA with* $\sigma = 0.5$*.*

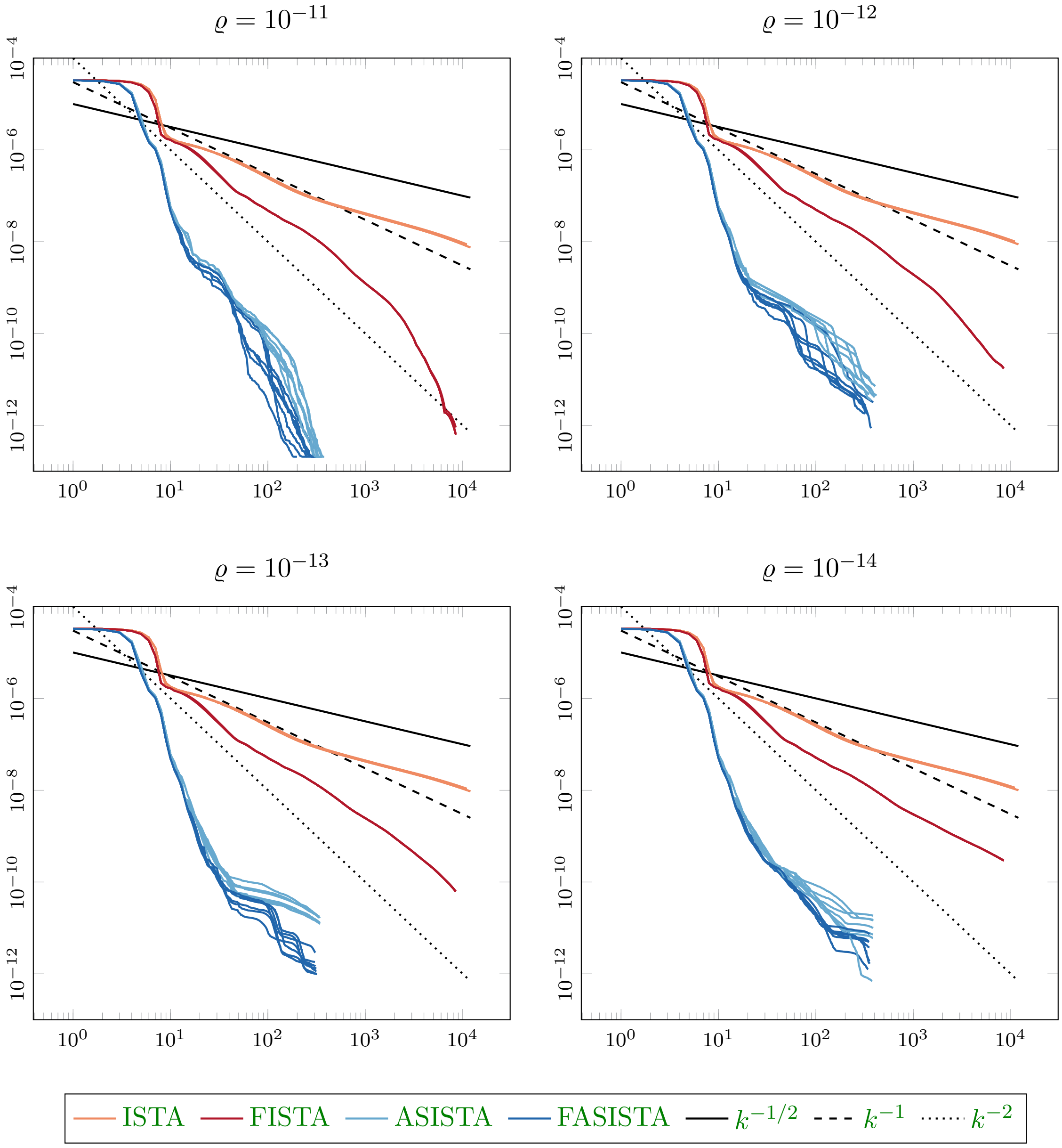

Figure 3. *First example: Distance to the estimated minimum (vertical axis), i.e. $J(\boldsymbol{b}_k) - J(\boldsymbol{b}^*)$, as a function of iteration number k (horizontal axis). The four plots show the four different regularisation strengths, $\varrho = 10^{-11}, 10^{-12}, 10^{-13}, 10^{-14}$. For each method, the different lines correspond to different choices for the greediness parameter, $\sigma = 0.3, 0.4, \ldots, 0.9$.*

Table 1

Relative L^2- and H^1-errors in the data for the different noise levels considered in the second example.

δ	$\frac{\|u_d-u_r\|_{L^2(\Omega)}}{\|u_r\|_{L^2(\Omega)}}$	$\frac{\|u_d-u_r\|_{H^1(\Omega)}}{\|u_r\|_{H^1(\Omega)}}$
$10^{-1.5}$	0.0316228	7.8774228
10^{-2}	0.0100000	2.4926186
$10^{-2.5}$	0.0031624	0.7889325
10^{-3}	0.0010003	0.2499083
$10^{-3.5}$	0.0003173	0.0802032
10^{-4}	0.0001032	0.0289778

The resulting reconstructions are shown in Figures 4 and 5. In both cases, higher noise in the data requires stronger regularisation to be able to sufficiently suppress noise in the reconstruction. The reconstructions using the discrete cosine expansion shown in Figure 5 suffer from a wrong reconstruction near the centre of the image. This likely happens due to an interplay between the expansion and the known difficulty of determining the coefficient near points where the gradient of the state u vanishes, see [17]. The reconstructions using the Haar wavelet expansion shown in Figure 4 seem to suffer less from this, with that area instead appearing more pixelated. Generally, the reconstructions in Figure 4 appear very pixelated for large regularisation strengths by nature of the Haar wavelets.

7. Conclusion. In this article, we considered the reconstruction of an unknown diffusion coefficient from measurements of the PDE solution inside the domain of interest. This ill-posed problem was stated as a nonlinear optimal control problem and regularised by sparsity constraints for the diffusion coefficient, which was represented by either a Haar wavelet expansion or a cosine series expansion. We investigated the functional analytic setup and determined variants of the ISTA and FISTA methods for the minimisation. Moreover, by a novel combination of known approaches we derived the minimisation methods ASISTA and FASISTA. So far for all these methods, we can only provide a heuristic line search. The numerical examples demonstrated that the sparsity constraints can be used to control noise in the reconstruction and indicated that the ASISTA and FASISTA methods are more efficient than the ISTA and FISTA methods. Finally, we would like to mention that the ASISTA and FASISTA methods might be able to be improved further by developing suitable preconditioning for the conjugate gradient method that is part of their inner iteration.

Acknowledgements. The authors thank Yannik Gleichmann for helpful and insightful discussions. The colour map utilised in the depictions is “CET-L04” from the perceptually uniform colour map collection “colorCET”, cf. [18], available at https://colorcet.com.

Data Availability. The numerical examples presented in this article can be replicated solely using the information contained in this article. In addition, the MATLAB code that computed the numerical examples is available as [11].

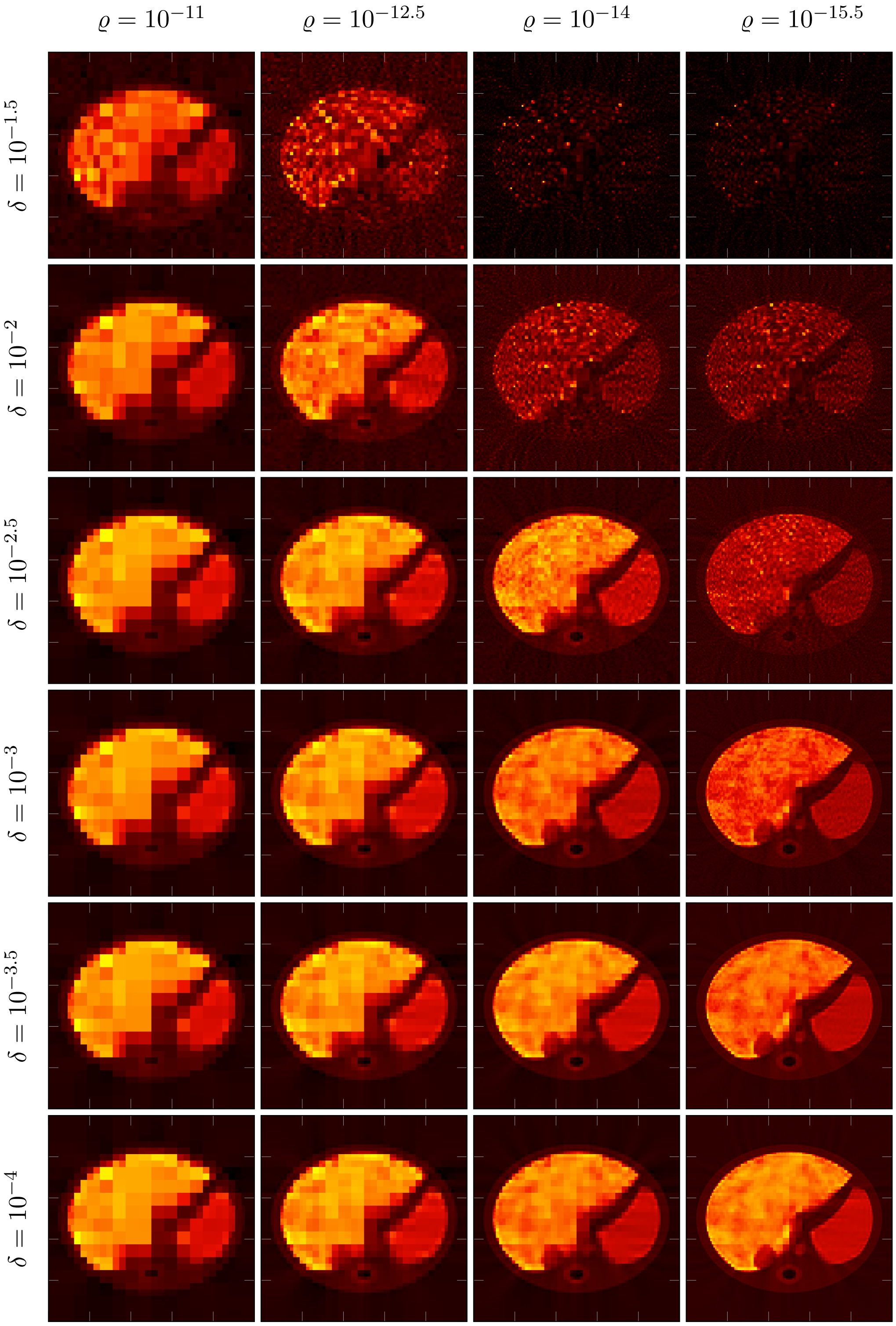

Figure 4. *Second example: Reconstructions using the Haar wavelet expansion. In each row, the regularisation strength decreases from left to right, while in each column the noise level decreases from top to bottom.*

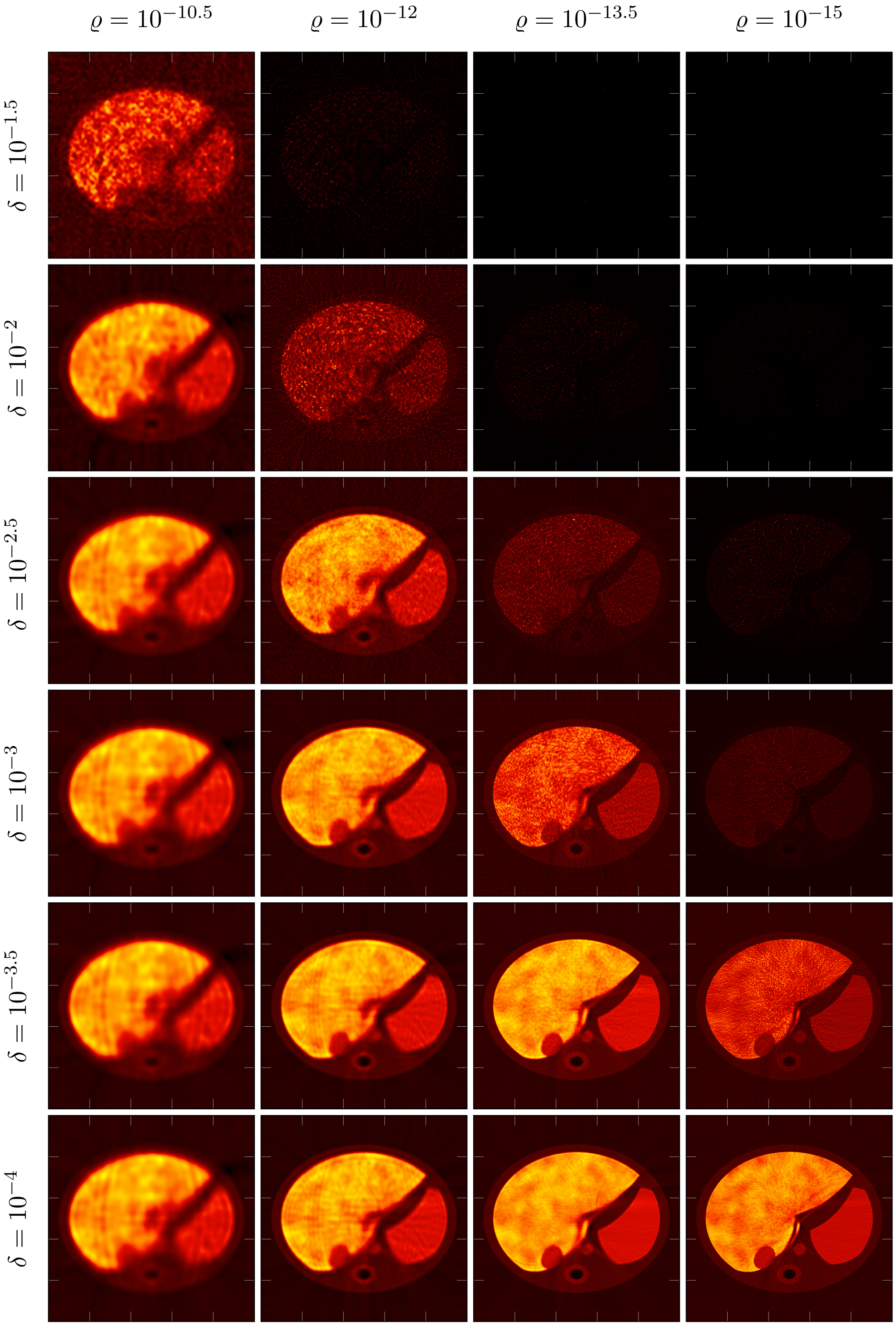

Figure 5. *Second example: Reconstructions using the discrete cosine expansion. In each row, the regularisation strength decreases from left to right, while in each column the noise level decreases from top to bottom.*

REFERENCES

[1] U. Assmann and A. Rösch, *Identification of an unknown parameter function in the main part of an elliptic partial differential equation*, J. Anal. Appl., 32 (2013), pp. 163–178.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2(1) (2009), pp. 183–202.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2010), p. 1–122.

[4] E. Candes and D. Donoho, *Curvelets: A surprisingly effective nonadaptive representation for objects with edges*, in Curves and Surface Fitting: Saint-Malo 1999, A. Cohen, C. Rabut, and L. Schumaker, eds., Nashville, 2000, Vanderbilt University Press, p. 105–120.

[5] X. Chen, Z. Nashed, and L. Qi, *Smoothing methods and semismooth methods for nondifferentiable operator equations*, SIAM J. Numer. Anal., 38 (2000), pp. 1200–1216.

[6] F. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[7] A. Cohen and R. DeVore, *Approximation of high-dimensional parametric PDEs*, Acta Numer., 24 (2015), pp. 1–159.

[8] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457.

[9] M. Do and V. M., *The contourlet transform: an efficient directional multiresolution image representation*, Image Process. IEEE Trans., 14 (2005), pp. 2091–2106.

[10] H. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, vol. 375 of Mathematics and Its Applications, Kluwer Academic Publishers Group, Dordrecht, 1996.

[11] L. Felber and M. Schmidlin, *Code for: Identification of sparsely representable diffusion parameters in elliptic problems*, Apr. 2023, https://doi.org/10.5281/zenodo.7821730.

[12] K. Glaser, A. Manduca, and R. Ehman, *Review of MR elastography applications and recent developments*, J. Magn. Reson. Imaging, 36 (2012), pp. 757–774.

[13] R. Griesse and D. Lorenz, *A semismooth Newton method for Tikhonov functionals with sparsity constraints*, Inverse Problems, 24 (2008), p. 035007.

[14] K. Guo and D. Labate, *Optimally sparse multidimensional representation using shearlets*, SIAM J. Math. Anal., 39 (2007), pp. 298–318.

[15] H. Harbrecht, *A finite element method for elliptic problems with stochastic input data*, Appl. Numer. Math., 60 (2010), pp. 227–244.

[16] A. Kirsch, *An Introduction to the Mathematical Theory of Inverse Problems*, vol. 120 of Applied Mathematical Sciences, Springer, New York, 2nd ed., 2011.

[17] I. Knowles, *Parameter identification for elliptic problems*, J. Comput. Appl. Math., 131 (2001), pp. 175–194.

[18] P. Kovesi, *Good colour maps: How to design them*, arXiv:1509.03700v1 [cs.GR], (2015).

[19] J. Lee, Y. Sun, and M. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. Optim., 24 (2014), pp. 1420–1443.

[20] D. Lorenz, P. Maass, and P. Muoi, *Gradient descent for Tikhonov functionals with sparsity constraints: Theory and numerical comparison of step size rules*, Electron. Trans. Numer. Anal., 39 (2012), pp. 437–463.

[21] A. Milzarek and M. Ulbrich, *A semismooth Newton method with multidimensional filter globalization for ℓ_1-optimization*, SIAM J. Optim., 24 (2014), pp. 298–333.

[22] P. Muoi, D. Háo, P. Maass, and M. Pidcock, *Semismooth Newton and quasi-Newton methods in weighted ℓ^1-regularization*, J. Inverse Ill-Posed Probl., 21 (2013), pp. 665–693.

[23] M. Murphy, D. Jones, C. Jack Jr, K. Glaser, M. Senjem, A. Manduca, J. Felmlee, R. Carter, R. Ehman, and J. Huston III, *Regional brain stiffness changes across the Alzheimer's disease spectrum*, NeuroImage: Clinical, 10 (2016), pp. 283–290.

[24] Y. Nesterov, *A method of solving a convex pogramming problem with convergence rate $\mathcal{O}(\frac{1}{k^2})$*, Dokl. Akad. Nauk SSSR, 269(3) (1983), pp. 543–547.

[25] J. Tausch and J. White, *Multiscale bases for the sparse representation of boundary integral operators on complex geometry*, SIAM J. Sci. Comput., 24 (2003), pp. 1610–1629.

[26] M. Yushchenko, M. Sarracanie, and N. Salameh, *Fast acquisition of propagating waves in humans with low-field MRI: Toward accessible MR elastography*, Sci. Adv., 8 (2022), pp. 1–11.