

A Semismooth Newton Method for Tikhonov Functionals with Sparsity Constraints

R Griesse¹ and D A Lorenz²

¹Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenbergerstraße 69, A-4040 Linz, Austria

²Zentrum für Technomathematik, University of Bremen, D-28334 Bremen, Germany

E-mail: dlorenz@math.uni-bremen.de, roland.griesse@oeaw.ac.at

Abstract. Minimization problems in ℓ^2 for Tikhonov functionals with sparsity constraints are considered. Sparsity of the solution is ensured by a weighted ℓ^1 penalty term. The necessary and sufficient condition for optimality is shown to be slantly differentiable (Newton differentiable), hence a semismooth Newton method is applicable. Local superlinear convergence of this method is proved. Numerical examples are provided which show that our method compares favorably with existing approaches.

AMS classification scheme numbers: 65J22, 90C53, 49N45

1 Introduction

In this work we consider the optimization problem

$$\text{Minimize } \frac{1}{2} \|Ku - f\|_{\mathcal{H}}^2 + \sum_{k=1}^{\infty} w_k |u_k| \quad \text{over } u \in \ell^2. \quad (1)$$

Here, $K : \ell^2 \rightarrow \mathcal{H}$ is a linear and injective operator mapping the sequence space ℓ^2 into a Hilbert space \mathcal{H} , $f \in \mathcal{H}$ and $w = \{w_k\}$ is a sequence satisfying $w_k \geq w_0 > 0$.

One well understood algorithm for the solution of (1) is the so-called iterated soft-thresholding for which convergence has been proven in [10], see also [2, 9]. While the iterated soft-thresholding is very easy to implement it converges very slow in practice (in fact the method converges linearly but with a constant very close to one [2]). Another well analyzed method is the iterated hard-thresholding which converges like $\mathcal{O}(n^{-1/2})$ [3] (i.e. even slower than the iterated soft-thresholding but practically it is faster in many cases).

In this article we derive an algorithm for which we prove local superlinear convergence in the infinite dimensional setting. Our algorithm is an active set, or semismooth Newton, method and hence, the analysis is based on the notion of slant differentiability [8, 16]. The semismooth Newton method is easily implementable as an

active set method. Numerical experiments show that the method is robust with respect to the choice of the initial value and that it compares favorably with existing approaches in terms of computation time.

The background for problems of type (1) is, for example, the attempt to solve the linear operator equation $Ku = f$ in an infinite-dimensional Hilbert space which models the connection between some quantity of interest u and some measurements f . Often, the measurements f contain noise which makes the direct inversion ill-posed and practically impossible. Thus, instead of considering the linear equation, a regularized problem is posed for which the solution is stable with respect to noise. A common approach is to regularize by minimizing a Tikhonov functional [10, 13, 21]. A special class of these regularizations has been of recent interest, namely of the type (1). These problems model the fact that the quantity of interest u is composed of a few elements, i.e. it is sparse in some given, countable basis. To make this precise, let $A : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ be a bounded operator between two Hilbert spaces and let $\{\psi_k\}$ be an orthonormal basis of \mathcal{H}_1 . Denote by $B : \ell^2 \rightarrow \mathcal{H}_1$ the synthesis operator $B(u_k) = \sum_k u_k \psi_k$. Then the problem

$$\min_{u \in \mathcal{H}_1} \frac{1}{2} \|Au - f\|_{\mathcal{H}_2}^2 + \sum_{k=1}^{\infty} w_k |\langle u, \psi_k \rangle|$$

can be rephrased as

$$\min_{u \in \ell^2} \frac{1}{2} \|ABu - f\|_{\mathcal{H}_2}^2 + \sum_{k=1}^{\infty} w_k |u_k|.$$

The sequence w_k plays the role of the regularization parameter where each coefficient is regularized individually. However, for an analysis of the regularizing properties one might use αw_k instead and investigate $\alpha \rightarrow 0$. We refer to e.g. [10, 18, 20] for analysis of the regularizing properties and parameter choice rules.

Recently sparsity constraints have also appeared in the context of optimal control of PDEs [24].

The article is organized as follows. In Section 2 we derive a semismooth formulation for the minimization problem (1). Section 3 states the algorithm and local superlinear convergence is proven. The Section 4 presents numerical results on the regularization of the ill-posed problems of inverse integration and deblurring and shows an application to ℓ^1 minimization in the context of compressed sensing.

Notation. For $1 \leq p < \infty$, ℓ^p denotes the space of p -summable sequences with norm $\|u\|_p = \left(\sum_{k=1}^{\infty} |u_k|^p \right)^{1/p}$, whereas ℓ^∞ denotes the space of bounded sequences with norm $\|u\|_\infty = \max_{k \in \mathbb{N}} |u_k|$. Recall that these spaces satisfy $\ell^p \hookrightarrow \ell^q$ for $1 \leq p \leq q \leq \infty$ and that $|u_k| \leq \|u\|_p$ holds for any $u \in \ell^p$. In the case $p = 2$ we simply write $\|u\|$, and $\langle \cdot, \cdot \rangle$ denotes the inner product in ℓ^2 . With $B_\rho(u)$ we denote the open ball of radius ρ with respect to the norm of ℓ^2 , centered at u . The operator $K^* : \mathcal{H} \rightarrow \ell^2$ is the Hilbert space adjoint of K and $L(X, Y)$ is the space of bounded linear operators from X to Y .

2 Optimality Conditions

In this section we are going to derive the necessary and sufficient optimality condition for the problem (1). It is going to be the basis for the semismooth Newton algorithm. This condition can be derived and expressed in different ways, for example by using the classical Lagrange duality, or by using subgradient calculus.

Let us first address the conditions obtained by subgradient calculus. To this end we introduce the so-called soft-thresholding function.

Definition 2.1. Let $w = \{w_k\}$ with $w_k \geq w_0 > 0$ and $1 \leq p < \infty$, $1 \leq q \leq \infty$. The soft-thresholding of u with the sequence w is defined as the mapping $\mathcal{S}_w : \ell^p \rightarrow \ell^q$ given by

$$\mathcal{S}_w(u)_k = S_{w_k}(u_k) = \max\{0, |u_k| - w_k\} \operatorname{sgn}(u_k). \quad (2)$$

Remark 2.2. Since elements of ℓ^p are sequences converging to zero, the range of \mathcal{S}_w is $\ell^0 = \{u \in \mathbb{R}^{\mathbb{N}} : u_k = 0 \text{ for almost every } k\} \subset \ell^q$.

With the help of the soft-thresholding operator, we can formulate the optimality condition in a compact way.

Proposition 2.3. If $K : \ell^2 \rightarrow \mathcal{H}$ is injective, the functional

$$\Psi(u) = \frac{1}{2} \|Ku - f\|_{\mathcal{H}}^2 + \sum_{k=1}^{\infty} w_k |u_k| \quad (3)$$

has a unique minimizer $\bar{u} \in \ell^2$. This minimizer is characterized by

$$\bar{u} = \mathcal{S}_{\gamma w}(\bar{u} - \gamma K^*(K\bar{u} - f)) \quad \text{for any } \gamma > 0. \quad (4)$$

Proof. Since K is injective, Ψ is strictly convex and coercive and hence, then it has a unique minimizer. This minimizer is characterized by

$$0 \in \partial\Psi(\bar{u})$$

which is equivalent to

$$-K^*(K\bar{u} - f) \in \partial F(\bar{u}) \quad (5)$$

where $F(u) = \sum_k w_k |u_k|$. Multiplying with $\gamma > 0$, adding \bar{u} to both sides and inverting $(I + \gamma \partial F)$ gives

$$\bar{u} = (I + \gamma \partial F)^{-1}(\bar{u} - \gamma K^*(K\bar{u} - f)).$$

(Note that $(I + \gamma \partial F)^{-1}$ exists and is single-valued since the subgradient ∂F is maximal monotone if F is convex and lower semicontinuous [26, Proposition 32.17, Corollary 32.30].) A straightforward calculation shows that

$$(I + \gamma \partial F)^{-1} = \mathcal{S}_{\gamma w}.$$

□

From the characterization (4) and Remark 2.2 we can derive the following corollary.

Corollary 2.4. *The minimizer \bar{u} of (3) is a finitely supported sequence.*

For convenience we also derive the optimality condition by Lagrange duality. We split the functional Ψ according to

$$\Psi(u) = G(Ku) + F(u)$$

with $G(h) = \|h - f\|_{\mathcal{H}}^2/2$ and $F(u) = \sum_k w_k |u_k|$. To state the dual problem, we use the dual variable p which shall not be confused with exponents for ℓ^p spaces. The dual variable appears only in this section. The dual problem of (1) is defined as

$$\text{Maximize} \quad -F^*(K^*p) - G^*(-p) \quad \text{over } p \in \mathcal{H}. \quad (6)$$

This can be expressed as (see the appendix)

$$\begin{aligned} \text{Maximize} \quad & -\frac{1}{2}\|p\|_{\mathcal{H}}^2 + \langle p, f \rangle \quad \text{over } p \in \mathcal{H} \\ \text{subject to} \quad & |K^*p|_k \leq w_k \quad \text{for all } k. \end{aligned}$$

The extremality conditions [12, Ch. III.4] are:

$$F(u) + F^*(K^*p) - \langle K^*p, u \rangle = 0 \quad (7a)$$

$$G(Ku) + G^*(-p) + \langle p, Ku \rangle = 0. \quad (7b)$$

The first condition (7a) yields

$$\begin{aligned} \sum_{k=1}^{\infty} \underbrace{w_k |u_k|}_{\geq 0} - \langle K^*p, u \rangle &= 0 \quad \text{and} \quad |(K^*p)_k| \leq w_k \\ \Leftrightarrow u_k = 0 \quad \text{or} \quad (K^*p)_k &= w_k \text{ sign } u_k = \begin{cases} w_k, & \text{if } u_k > 0 \\ -w_k, & \text{if } u_k < 0 \end{cases} \\ \text{and} \quad |(K^*p)_k| &\leq w_k. \end{aligned}$$

This condition can be written as the complementarity system

$$\begin{aligned} K^*p - w &\leq 0, & u^+ &\geq 0, & [K^*p - w] u^+ &= 0 \\ -K^*p - w &\leq 0, & u^- &\geq 0, & [K^*p + w] u^- &= 0, \end{aligned} \quad (8)$$

in a coordinatewise sense, which is in turn equivalent to

$$u = \max\{0, u + \gamma(K^*p - w)\} + \min\{0, u + \gamma(K^*p + w)\} \quad (9)$$

for any $\gamma > 0$.

The second condition (7b) yields

$$\frac{1}{2}\|Ku - f\|_{\mathcal{H}}^2 + \frac{1}{2}\|-p\|_{\mathcal{H}}^2 + \langle -p, f \rangle + \langle p, Ku \rangle = \frac{1}{2}\|Ku - f + p\|_{\mathcal{H}}^2 = 0$$

and thus

$$Ku - f + p = 0. \quad (10)$$

By plugging (10) into (9) we end up with

$$u - \max\{0, u - \gamma(K^*(Ku - f) + w)\} - \min\{0, u - \gamma(K^*(Ku - f) - w)\} = 0, \quad (11)$$

which is just another way to express (4).

Remark 2.5. *The usual characterization $0 \in \partial\Psi(\bar{u})$ of the unique minimizer \bar{u} of (1) is difficult to handle for numerical algorithms because it is a nonsmooth inclusion. One attempt to tackle the problem is by interior point regularization as proposed in [17]. This, however, introduces additional nonlinearities into the problem. By contrast, our algorithm is based on the necessary and sufficient condition (11). As we shall prove in the following section, (11) is a semismooth equation in ℓ^2 , so that Newton's method can be applied.*

3 Semismooth Newton Method

The previous section has shown that we can solve the minimization problem (1) by solving the equation (4) or (11), or briefly

$$\mathcal{F}(u) = u - \mathcal{S}_{\gamma w}(u - \gamma K^*(Ku - f)) = 0, \quad (12)$$

for some $\gamma > 0$.

This is an operator equation in the space ℓ^2 , involving the non-differentiable max and min operations. Optimality conditions of this form frequently also occur in the context of optimal control problems for partial differential equations, in the presence of control constraints. Then (12) is considered in L^p function spaces, and it is known that the max operation, i.e., $u \mapsto \max\{0, u\}$, is so-called Newton or slantly differentiable from L^p to L^q for $1 \leq q \leq p \leq \infty$, see [8, Theorem 2.6] in view of its Lipschitz continuity. In the presence of a norm gap $1 \leq q < p \leq \infty$, the generalized derivative, or slanting function, can be chosen as an indicator function, see [16, Proposition 4.1]. This allows for the interpretation of the generalized Newton method as a so-called active set method. This norm gap is made up for in the context of partial differential equation because K and K^* are solution operators which provide the necessary smoothing.

It turns out that the behavior of the max and min operations is more intricate than in function space. Again, it follows from the Lipschitz continuity of $u \mapsto \max\{0, u\}$ from ℓ^p to ℓ^q that slant differentiability holds [8, Theorem 2.6] for $1 \leq p, q \leq \infty$. However, we are not aware of any simple slanting function even with norm gap which can be algorithmically exploited, see Remark 3.2. It may be surprising that nonetheless, the soft-thresholding operator \mathcal{S}_w and thus equation (12) are slantly differentiable and admit a simple slanting function between *any* pair of ℓ^p , ℓ^q spaces, see Proposition 3.3. This allows us to apply a generalized Newton's method to solve (12), which takes the form of an active set method.

3.1. Semismoothness of the optimality condition. The concept of slant or Newton differentiability is closely related to the notion of semismoothness [8, 16, 25], and we will use the terms interchangeably.

Definition 3.1. Let X and Y be Banach spaces and $D \subset X$ be an open subset. A mapping $\mathcal{F} : D \rightarrow Y$ is called Newton (or slantly) differentiable in $x \in D$ if there exists a family of mappings $\mathcal{G} : D \rightarrow L(X, Y)$ such that

$$\lim_{h \rightarrow 0} \frac{\|\mathcal{F}(x+h) - \mathcal{F}(x) - \mathcal{G}(x+h)h\|_Y}{\|h\|_X} = 0. \quad (13)$$

The function \mathcal{G} is called a generalized derivative (or slanting function) for \mathcal{F} in x .

It is shown in [8] that any Lipschitz continuous function is Newton differentiable. However, this is only of little help algorithmically unless there is a generalized derivative $\mathcal{G}(u)$ of (12) which is easily invertible.

Remark 3.2. A natural candidate for a generalized derivative \mathcal{G} of the function $\mathcal{F}(u) = \max(0, u)$ is

$$\mathcal{G}(u)(h)_k = \begin{cases} h_k & , u_k > 0 \\ \delta h_k & , u_k = 0 \\ 0 & , u_k < 0 \end{cases} \quad \text{for any } \delta \in \mathbb{R}.$$

We are going to show that this \mathcal{G} can not serve as a generalized derivative of $\mathcal{F} : \ell^p \rightarrow \ell^q$ for any $p \in [1, \infty[$ and $1 \leq q \leq \infty$. We consider a point $u \in \ell^p$ for which the set $\{n \mid u_n \neq 0\}$ is infinite and take a special sequence of $h^n \in \ell^p$, namely

$$h_k^n = \begin{cases} 0 & \text{for } k \neq n \\ -2u_k & \text{for } k = n. \end{cases}$$

Hence, we have $\|h^n\|_p = 2|u_n| \rightarrow 0$ for $n \rightarrow \infty$. It is an easy calculation to see that

$$\frac{\|\max\{u + h^n, 0\} - \max\{u, 0\} - G(u + h^n)h^n\|_q}{\|h^n\|_p} = \frac{1}{2} \quad \text{for all } n \text{ with } u_n \neq 0.$$

The following proposition shows that the thresholding operator (2) is Newton differentiable and that a function similar to \mathcal{G} serves as a generalized derivative.

Proposition 3.3. The mapping $\mathcal{S}_w : \ell^p \rightarrow \ell^q$ from Definition 2.1 is Newton differentiable for any $1 \leq p < \infty$, $1 \leq q \leq \infty$. A generalized derivative is given by

$$(\mathcal{G}(u)v)_k = \begin{cases} v_k & \text{for } |u_k| > w_k \\ 0 & \text{for } |u_k| \leq w_k. \end{cases}$$

Proof. Without loss of generality we may assume $\|h\|_p < \frac{w_0}{2}$ and hence $|h_k| < \frac{w_0}{2}$. Since $u \in \ell^p$ with $p < \infty$ there exists k_0 such that $|u_k| < \frac{w_0}{2}$ for $k > k_0$. We estimate

$$\begin{aligned} & \|\mathcal{S}_w(u+h) - \mathcal{S}_w(u) - \mathcal{G}(u+h)(h)\|_q^q \\ &= \sum_{k=1}^{\infty} |S_{w_k}(u_k+h_k) - S_{w_k}(u_k) - \mathcal{G}(u+h)(h)_k|^q \\ &= \sum_{\substack{k \leq k_0 \\ |u_k| \neq w_k}} |S_{w_k}(u_k+h_k) - S_{w_k}(u_k) - \mathcal{G}(u+h)(h)_k|^q. \end{aligned}$$

It is easy to check that the above sum is zero for

$$\|h\|_p < \min\{|u_k| - w_k| : k \leq k_0 \text{ and } |u_k| \neq w_k\}$$

because $|h_k| \leq \|h\|_p$ holds. It follows that

$$\frac{\|\mathcal{S}_w(u+h) - \mathcal{S}_w(u) - \mathcal{G}(u+h)(h)\|_q}{\|h\|_p} = 0$$

for $\|h\|_p$ small enough, which proves Newton differentiability. \square

Remark 3.4. In matrix notation we can express the generalized derivative $\mathcal{G}(u)$ as

$$\mathcal{G}(u) = \begin{pmatrix} I_{\mathcal{A}} & 0 \\ 0 & 0 \end{pmatrix}$$

where $\mathcal{A} = \{k \in \mathbb{N} : |u_k| > w_k\}$.

To calculate a generalized derivative for the mapping \mathcal{F} in (12), we prove a chain rule for the generalized derivative.

Lemma 3.5. Let $S : X \rightarrow Y$ be Newton differentiable, $A \in L(X, X)$ and $y \in X$. Let furthermore \mathcal{G} be a generalized derivative of S . Define $T(u) = S(Au + y)$. Then $H(u) = \mathcal{G}(Au + y)A$ is a generalized derivative of T .

Proof. It holds

$$\begin{aligned} & \frac{\|T(u+h) - T(u) - H(u+h)h\|}{\|h\|} \\ &= \frac{\|S(Au + Ah + y) - S(Au + y) - \mathcal{G}(Au + Ah + y)Ah\|}{\|Ah\|} \frac{\|Ah\|}{\|h\|}. \end{aligned}$$

The right hand side converges to zero because \mathcal{G} is a generalized derivative of S in $Au + y$ in the direction Ah , and A is bounded. \square

In order to specify a generalized derivative of \mathcal{F} , we introduce the active and the inactive sets. For the sake of simplicity we will restrict ourself to the case $\mathcal{F} : \ell^2 \rightarrow \ell^2$ in the following

Definition 3.6. For $u \in \ell^2$, the active set $\mathcal{A}(u)$ and the inactive set $\mathcal{I}(u)$ are given by

$$\begin{aligned}\mathcal{A}(u) &= \{k \in \mathbb{N} : |u - \gamma K^*(Ku - f)|_k > \gamma w_k\} \\ \mathcal{I}(u) &= \{k \in \mathbb{N} : |u - \gamma K^*(Ku - f)|_k \leq \gamma w_k\}.\end{aligned}$$

Whenever the active and inactive sets correspond to an iterate u^n , we will denote them by \mathcal{A}_n and \mathcal{I}_n , respectively. We will drop the subscript or the argument if no ambiguity can occur.

We are now in the position to calculate a generalized derivative of \mathcal{F} .

Proposition 3.7. The mapping $\mathcal{F} : \ell^2 \rightarrow \ell^2$,

$$\mathcal{F}(u) = u - \mathcal{S}_{\gamma w}(u - \gamma K^*(Ku - f))$$

is Newton differentiable. Denote the active and inactive set \mathcal{A} and \mathcal{I} as in Definition 3.6 and split the operator K^*K according to

$$K^*K = \begin{pmatrix} \mathcal{M}_{\mathcal{A}\mathcal{A}} & \mathcal{M}_{\mathcal{A}\mathcal{I}} \\ \mathcal{M}_{\mathcal{I}\mathcal{A}} & \mathcal{M}_{\mathcal{I}\mathcal{I}} \end{pmatrix}.$$

Then a generalized derivative is given by

$$\mathcal{G}(u) = \begin{pmatrix} 0 & 0 \\ 0 & I_{\mathcal{I}} \end{pmatrix} + \begin{pmatrix} I_{\mathcal{A}} & 0 \\ 0 & 0 \end{pmatrix} (\gamma K^*K) = \begin{pmatrix} \gamma \mathcal{M}_{\mathcal{A}\mathcal{A}} & \gamma \mathcal{M}_{\mathcal{A}\mathcal{I}} \\ 0 & I_{\mathcal{I}} \end{pmatrix}. \quad (14)$$

Proof. The claim follows from the sum rule for the generalized derivative and from Proposition 3.3 and Lemma 3.5 with $S = \mathcal{S}_{\gamma w}$, $A = I - \gamma K^*K$, $y = \gamma K^*f$. \square

Remark 3.8. Note that for any $u \in \ell^2$, the active set \mathcal{A} is always finite, since $u - \gamma K^*(Ku - f) \in \ell^2$ holds and thus $|u - \gamma K^*(Ku - f)|_k \rightarrow 0$ for $k \rightarrow \infty$.

3.2. Semismooth Newton method. The semismooth or generalized Newton method for the solution of (12) can be stated as the iteration

$$u^{n+1} = u^n - \mathcal{G}(u^n)^{-1} \mathcal{F}(u^n), \quad (15)$$

where \mathcal{G} is a generalized derivative of \mathcal{F} . We use the generalized derivative \mathcal{G} given by (14). Naturally, the semismooth Newton method can be interpreted as an active set method, and we state it as Algorithm 1.

Remark 3.9. (i) Algorithm 1 is the generalized Newton method (15). The unique solvability in step 8 is shown in Proposition 3.11 below.

(ii) Given an initial iterate $u^0 \in \ell^2$, the algorithm is well-defined, and all iterates remain in ℓ^2 . We refer again to Proposition 3.11 below.

(iii) At the end of step 10, the iterate u^{n+1} satisfies $u_{\mathcal{I}_n}^{n+1} = 0$. Note that $r_{\mathcal{I}_n}^n = u_{\mathcal{I}_n}^n$ holds which implies $\delta u_{\mathcal{I}_n} = -u_{\mathcal{I}_n}^n$.

Note that (iii) implies that all iterates u^n ($n \geq 1$) of Algorithm 1 are finitely supported sequences. However, $K^*(Ku^n - f)$ is in general not finitely supported, and hence in a practical implementation, this term will be truncated after a number of entries.

Algorithm 1 Semismooth Newton method for the solution of (12).

- 1: Initialize u^0 , choose $\gamma > 0$, set $n := 0$ and $\text{done} := \text{false}$
- 2: **while** $n < n_{max}$ and not done **do**
- 3: Calculate the active and inactive sets:

$$\begin{aligned}\mathcal{A} &= \{k \in \mathbb{N} : |u^n - \gamma K^*(Ku^n - f)|_k > \gamma w_k\} \\ \mathcal{I} &= \{k \in \mathbb{N} : |u^n - \gamma K^*(Ku^n - f)|_k \leq \gamma w_k\}.\end{aligned}$$

- 4: Compute the residual

$$r^n = \mathcal{F}(u^n) = u^n - \mathcal{S}_{\gamma w}(u^n - \gamma K^*(Ku^n - f)).$$

- 5: **if** $\|r^n\| \leq \varepsilon$ **then**
- 6: $\text{done} := \text{true}$
- 7: **else**
- 8: Calculate the Newton update by solving

$$\begin{pmatrix} \gamma \mathcal{M}_{\mathcal{A}\mathcal{A}} & \gamma \mathcal{M}_{\mathcal{A}\mathcal{I}} \\ 0 & I_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} \delta u_{\mathcal{A}} \\ \delta u_{\mathcal{I}} \end{pmatrix} = - \begin{pmatrix} r_{\mathcal{A}}^n \\ r_{\mathcal{I}}^n \end{pmatrix}$$

- 9: Update $u^{n+1} := u^n + \delta u$
 - 10: Set $n := n + 1$
 - 11: **end if**
 - 12: **end while**
-

3.3. Active set method. One may set up Algorithm 1 equivalently as an active set method. This can be seen by a closer analysis of the Newton step (step 8 and 9 in Algorithm 1):

$$\begin{aligned}u^{n+1} &= u^n - \begin{pmatrix} \frac{1}{\gamma} \mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1} & -\mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1} \mathcal{M}_{\mathcal{A}\mathcal{I}} \\ 0 & I_{\mathcal{I}} \end{pmatrix} \left(u^n - \mathcal{S}_{\gamma w}(u^n - \gamma K^*(Ku^n - f)) \right) \\ &= u^n - \begin{pmatrix} \frac{1}{\gamma} \mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1} & -\mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1} \mathcal{M}_{\mathcal{A}\mathcal{I}} \\ 0 & I_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} \gamma [K^*(Ku^n - f)]_{\mathcal{A}} \pm w_{\mathcal{A}} \\ u_{\mathcal{I}}^n \end{pmatrix} \\ &= \begin{pmatrix} u_{\mathcal{A}}^n - \mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1} ([K^*(Ku^n - f)]_{\mathcal{A}} \pm w_{\mathcal{A}} - \mathcal{M}_{\mathcal{A}\mathcal{I}} u_{\mathcal{I}}^n) \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1} (K^* f \pm w)|_{\mathcal{A}} \\ 0 \end{pmatrix}\end{aligned}$$

The sign of w depends of the sign of $u^n - \gamma K^*(Ku^n - f)$. Hence, instead of calculating the Newton update in step 8, one may set $u_{\mathcal{I}}^{n+1} := 0$ and solve $\mathcal{M}_{\mathcal{A}\mathcal{A}} u_{\mathcal{A}}^{n+1} = (K^* f \pm w)|_{\mathcal{A}}$. This shows that the subsequent iterate u^{n+1} depends on the current iterate u^n solely through the active set \mathcal{A} . As a consequence, differing values of u^n can lead to the same

next iterate u^{n+1} .

For completeness, we state the active set method as Algorithm 2. Note that the algorithm is initialized with an active set \mathcal{A} instead of u^0 .

Algorithm 2 Active set method for the solution of (12).

- 1: Initialize \mathcal{A}_0^+ , \mathcal{A}_0^- , choose $\gamma > 0$, set $n := 0$ and `done := false`
- 2: Set $\mathcal{A}_0 = \mathcal{A}_0^+ \cup \mathcal{A}_0^-$, $\mathcal{I}_0 = \mathbb{N} \setminus \mathcal{A}_0$
- 3: Set the signs of the weights:

$$s_k^0 = \begin{cases} 1, & k \in \mathcal{A}_0^+ \\ 0, & k \in \mathcal{I}_0 \\ -1, & k \in \mathcal{A}_0^- \end{cases}$$

- 4: **while** $n < n_{max}$ and not `done` **do**
- 5: Set $u_{\mathcal{I}_n}^n = 0$ and calculate $u_{\mathcal{A}_n}^n$ by solving

$$\mathcal{M}_{\mathcal{A}_n \mathcal{A}_n} u_{\mathcal{A}_n}^n = (K^* f + s^n w)|_{\mathcal{A}_n}$$

- 6: Calculate the new active sets:

$$\begin{aligned} \mathcal{A}_{n+1}^+ &= \{k \in \mathbb{N} : [u^n - \gamma K^*(Ku^n - f)]_k > \gamma w_k\} \\ \mathcal{A}_{n+1}^- &= \{k \in \mathbb{N} : [u^n - \gamma K^*(Ku^n - f)]_k < -\gamma w_k\} \\ \mathcal{I}_{n+1} &= \{k \in \mathbb{N} : |u^n - \gamma K^*(Ku^n - f)|_k \leq \gamma w_k\}. \end{aligned}$$

- 7: Set the signs of the weights:

$$s_k^{n+1} = \begin{cases} 1, & k \in \mathcal{A}_{n+1}^+ \\ 0, & k \in \mathcal{I}_{n+1} \\ -1, & k \in \mathcal{A}_{n+1}^- \end{cases}$$

- 8: **if** $s^{n+1} = s^n$ **then**
 - 9: `done := true`
 - 10: **end if**
 - 11: Set $n := n + 1$
 - 12: **end while**
-

In this setting, the stopping criterion is coincidence of the active sets in consecutive iterations—other choices are also possible. In the numerical examples in Section 4 we chose the norm of the residual because a sudden drop of the residual norm occurred before the minimizer was identified.

3.4. Local convergence of the semismooth Newton method. The local superlinear convergence of the semismooth Newton method (Algorithm 1) hinges upon

the uniform boundedness of $\mathcal{G}(u^n)^{-1}$ during the iteration.

Proposition 3.10. *There exists $k_0 \in \mathbb{N}$ and $\rho > 0$ such that $\|u - \bar{u}\|_2 < \rho$ implies that*

$$\mathcal{A}(u) \subset [1, k_0].$$

Moreover, k_0 and ρ depend only on γ , \bar{u} , $\|K^*K\|$, $\|K^*f\|$, and w_0 .

Proof. The triangle inequality implies

$$|u - \gamma K^*(Ku - f)|_k \leq |\bar{u} - \gamma K^*(K\bar{u} - f)|_k + |u - \bar{u} - \gamma K^*K(u - \bar{u})|_k. \quad (16)$$

The first term can be estimated by

$$|\bar{u} - \gamma K^*(K\bar{u} - f)|_k \leq |\bar{u}|_k + \gamma |K^*K\bar{u}|_k + \gamma |K^*f|_k.$$

Since \bar{u} , $K^*K\bar{u}$ and K^*f are in ℓ^2 , the right hand side converges to 0 as $k \rightarrow \infty$. In particular, there exists k_0 , depending only on the named quantities, such that

$$|\bar{u} - \gamma K^*(K\bar{u} - f)|_k \leq \gamma w_0/2 \quad \text{for all } k \geq k_0. \quad (17)$$

The second term in (16) can be estimated by

$$\begin{aligned} |u - \bar{u} - \gamma K^*K(u - \bar{u})|_k &\leq |u - \bar{u}|_k + \gamma |K^*K(u - \bar{u})|_k \\ &\leq \|u - \bar{u}\| + \gamma \|K^*K(u - \bar{u})\| \leq (1 + \gamma \|K^*K\|) \|u - \bar{u}\|. \end{aligned}$$

Hence there exists $\rho > 0$, depending only on the named quantities, such that

$$|u - \bar{u} - \gamma K^*K(u - \bar{u})|_k \leq \gamma w_0/2 \quad \text{for all } k \in \mathbb{N}. \quad (18)$$

Combining (16)–(18) proves the claim. \square

At this point, we cannot yet conclude that the active sets remain uniformly bounded during the iteration of Algorithm 1, since it is not evident whether the iterates will remain in a suitable ρ -neighborhood of \bar{u} .

Proposition 3.11. *The generalized derivative \mathcal{G} , given by (14), is boundedly invertible from ℓ^2 into ℓ^2 . Moreover, the norm of $\mathcal{G}(u)^{-1}$ can be estimated by*

$$\|\mathcal{G}(u)^{-1}\| \leq \|\mathcal{M}_{\mathcal{A}\mathcal{A}}^{-1}\| \left(\frac{1}{\gamma} + \|\mathcal{M}_{\mathcal{A}\mathcal{I}}\| \right) + 1,$$

where \mathcal{A} and \mathcal{I} are the active and inactive sets at u , see Definition 3.6.

Proof. Let $u, r \in \ell^2$ and consider the equation $\mathcal{G}(u) \delta u = r$, i.e.,

$$\begin{pmatrix} \gamma \mathcal{M}_{\mathcal{A}\mathcal{A}} & \gamma \mathcal{M}_{\mathcal{A}\mathcal{I}} \\ 0 & I_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} \delta u_{\mathcal{A}} \\ \delta u_{\mathcal{I}} \end{pmatrix} = \begin{pmatrix} r_{\mathcal{A}} \\ r_{\mathcal{I}} \end{pmatrix}.$$

Necessarily, $\delta u_{\mathcal{I}} = r_{\mathcal{I}}$ holds, which implies $\delta u_{\mathcal{I}} \in \ell^2$. It remains to solve

$$\gamma \mathcal{M}_{\mathcal{AA}} \delta u_{\mathcal{A}} = r_{\mathcal{A}} - \gamma \mathcal{M}_{\mathcal{AI}} \delta u_{\mathcal{I}}. \quad (19)$$

The right hand side is an element of ℓ^2 . Moreover, $\mathcal{M}_{\mathcal{AA}}$ is injective. (We rewrite $\mathcal{M}_{\mathcal{AA}} = P_{\mathcal{A}} K^* K P_{\mathcal{A}} = (K P_{\mathcal{A}})^* K P_{\mathcal{A}}$, where $P_{\mathcal{A}}$ the projection of ℓ^2 onto the active set. Then $\mathcal{M}_{\mathcal{AA}} u = 0$ implies $\|K P_{\mathcal{A}} u\|^2 = \langle u, \mathcal{M}_{\mathcal{AA}} u \rangle = 0$, and hence $u_{\mathcal{A}} = 0$ since K is injective.) By Remark 3.8, the active set is finite, and thus $\mathcal{M}_{\mathcal{AA}}$ is an injective operator on a finite dimensional space, hence it is also surjective. We conclude that (19) has a unique solution $\delta u_{\mathcal{A}} \in \ell^2$, hence $\mathcal{G}(u)^{-1} : \ell^2 \rightarrow \ell^2$ exists.

The norm estimate follows from

$$\begin{aligned} \|\mathcal{G}(u)^{-1} r\| &= \left\| \begin{pmatrix} \frac{1}{\gamma} \mathcal{M}_{\mathcal{AA}}^{-1} & -\mathcal{M}_{\mathcal{AA}}^{-1} \mathcal{M}_{\mathcal{AI}} \\ 0 & I_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} r_{\mathcal{A}} \\ r_{\mathcal{I}} \end{pmatrix} \right\| \\ &\leq \frac{1}{\gamma} \|\mathcal{M}_{\mathcal{AA}}^{-1}\| \|r_{\mathcal{A}}\| + \|\mathcal{M}_{\mathcal{AA}}^{-1}\| \|\mathcal{M}_{\mathcal{AI}}\| \|r_{\mathcal{I}}\| + \|r_{\mathcal{I}}\| \\ &\leq \left(\|\mathcal{M}_{\mathcal{AA}}^{-1}\| \left(\frac{1}{\gamma} + \|\mathcal{M}_{\mathcal{AI}}\| \right) + 1 \right) \|r\|. \end{aligned}$$

□

Corollary 3.12. *Let $k_0 \in \mathbb{N}$ and $\rho > 0$ be as in Proposition 3.10. Then $\mathcal{G}(u)^{-1}$ is uniformly bounded on $B_{\rho}(\bar{u})$.*

Proof. Let $u \in \ell^2$ such that $\|u - \bar{u}\| < \rho$. By Proposition 3.10, the active set satisfies $\mathcal{A}(u) \subset [1, k_0]$. Our plan is to show that $\|\mathcal{G}(u)^{-1}\|$ indeed depends only on k_0 . Indeed, we define

$$C(k_0) := \max_{\emptyset \neq \mathcal{A} \subset [1, k_0]} \|\mathcal{M}_{\mathcal{AA}}^{-1}\| > 0.$$

Note that for every $\mathcal{A} \subset [1, k_0]$, $\mathcal{A} \neq \emptyset$, $\mathcal{M}_{\mathcal{AA}}$ is boundedly invertible, hence $C(k_0)$ is the maximum of finitely many positive numbers. Moreover, $\mathcal{M}_{\mathcal{AI}}$ is obtained from $K^* K$ by restriction and extension, hence $\|\mathcal{M}_{\mathcal{AI}}\| \leq \|K^* K\|$ holds, for all choices of \mathcal{A} and \mathcal{I} . From Proposition 3.11, we conclude that

$$\|\mathcal{G}(u)^{-1}\| \leq C(k_0) \left(\frac{1}{\gamma} + \|K^* K\| \right) + 1.$$

□

We may now combine the results above to argue the local superlinear convergence of Algorithm 1.

Theorem 3.13. *There exists a radius $r \in (0, \rho]$ such that $\|u^0 - \bar{u}\| < r$ implies that all iterates of Algorithm 1 satisfy $\|u^n - \bar{u}\| < r$, and $u^n \rightarrow \bar{u}$ superlinearly.*

Proof. By Corollary 3.12, the inverse of the generalized derivative, $\mathcal{G}(u)^{-1}$, remains uniformly bounded in $B_{\rho}(\bar{u})$. The result is then a standard conclusion for generalized Newton methods, see [7, Remark 2.7], or [16, Theorem 1.1]. □

Remark 3.14. (i) *The neighborhood in which superlinear convergence occurs is unknown and may be small. The global convergence behavior of the algorithm thus deserves further investigation. The numerical experiments in the following section suggest that the choice of γ is essential in achieving convergence from a bad initial guess. For a related problem in Hilbert spaces with a standard Tikhonov regularization term $\|u\|^2$, global convergence without rates was proved in [22].*

(ii) *The proof of Proposition 3.3 together with the chain rule (Lemma 3.5) shows that the remainder*

$$\mathcal{F}(u^n) - \mathcal{F}(\bar{u}) - \mathcal{G}(\bar{u})(u^n - \bar{u})$$

is exactly zero for sufficiently small $\|u^n - \bar{u}\|$. Hence we expect convergence in one step sufficiently close to the solution, which is confirmed by the numerical results in the following section.

Remark 3.15. *The assumption on the injectivity of K may be relaxed. The proof of Corollary 3.12 shows that we only need that all submatrices $\mathcal{M}_{\mathcal{A}\mathcal{A}}$ for $\mathcal{A} \subset [1, k_0]$ are invertible. Hence, local superlinear convergence can also be proved when the K satisfies the finite basis injectivity (FBI) property [2]. The FBI property states that any submatrix of K consisting of a finite number of columns is injective. The FBI property is related to the so-called restricted isometry property (RIP), see e.g. [1], which plays an important role in the analysis of minimizers of ℓ^1 constrained problems in the theory of compressed sensing [5].*

4 Numerical Results

In this section we present results of numerical experiments illustrating the performance of the semismooth Newton (SSN) method. We implemented the SSN method in MATLAB[®] and made experiments on a desktop PC with an AMD Athlon[™] 64 X2. Moreover, we are going to compare the SSN method to other state-of-the-art methods for the minimization of ℓ^1 constrained problems, namely the GPSR methods [14] and the 11_1s toolbox [17] where we used the freely available MATLAB[®] implementations of these methods. The GPSR method is based on gradient projection method with Barzilai-Borwein stepsizes and is known to converge r -linearly. The 11_1s method is a truncated Newton interior point method which is applied directly to the objective functional (note that we apply a Newton method to a reformulated optimality condition). In addition we included the widely used iterative soft-thresholding from [10] in our comparison. Note that both the GPSR and the 11_1s methods are set up and analyzed in a finite dimensional setting while our analysis on the SSN as well as the analysis for the iterative soft-thresholding is infinite dimensional.

4.1. Inverse integration. The problem under consideration is the classical ill-posed problem of inverse integration (or differentiation [3, 15, 23]), i.e. the operator K :

$L^2([0, 1]) \rightarrow L^2([0, 1])$ given by

$$Ku(t) = \int_0^t u(s) ds, \quad t \in [0, 1].$$

The data f is given as $(f(t_k))_{k=1, \dots, N}$ with $t_k = \frac{1}{N}k$. We discretized the operator K by the matrix

$$K = \frac{1}{N} \begin{pmatrix} 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \dots & \dots & 1 \end{pmatrix}, \quad K : \mathbb{R}^N \rightarrow \mathbb{R}^N.$$

The minimization problem reads

$$\min_{u \in \mathbb{R}^N} \frac{1}{2} \sum_{i=1}^N ((Ku)_i - f_i^\delta)^2 + \sum_{k=1}^N w_k |u_k|. \quad (20)$$

One can check easily that the SSN method is also applicable in finite dimensions and hence, this minimization problem can be treated by the SSN method. The discussion of the SSN method in infinite dimension provides us with results which are independent of the dimension N , i.e. the algorithm scales well.

The true solution \bar{u} is given by small plateaus and hence the data $f^\delta = K\bar{u} + \delta$ is a noisy function with steep linear ramps. Figure 1 shows our sample data and the result of the ℓ^1 minimization with the SSN method. Table 1 shows how the SSN method performed in this specific example. It can be observed that the residual is not decaying monotonically and it descends slowly in the beginning while it drops significantly in the last step. Moreover we observed in many examples that the algorithm shows a similar performance for a broad range of starting values u^0 . Another important observation is that the performance of the algorithm depends on the value of γ . For too small as well as for too large values of γ the algorithm does only converge when started very close to the solution. As a rule of thumb one could take γ close to the reciprocal of the smallest singular value of the (in practice unknown) matrix $\mathcal{M}_{\mathcal{A}\mathcal{A}}$ where \mathcal{A} is the sparsity pattern of the solution.

We made experiments to see how the SSN method depend on the noise level and the regularization parameter. First, we fixed the regularization sequence and changed the noise level. Hence, we solved the problem (20) for fixed $N = 500$, fixed $w_k = 10^{-5}$ and varied the noise level δ . Table 2 reports the results. Basically, a higher noise level leads to a smaller number of iterations but longer CPU-time (this is, because the active sets are larger during the iteration). Second we coupled the noise level and the regularizing sequence w_k . Since it is shown in [10, 18] that a parameter choice $w_k \propto \delta$ provides a regularization we used $w_k = \delta$. Hence, we solved the problem (20) for fixed $N = 500$ and different noise levels δ , see Table 3 for the results. Basically, the algorithm behaves similar for different noise levels, especially the CPU-time is always comparably small.

Moreover, we made a simple experiment to assess how the computational cost grow with the size of the problem. We considered the inverse intergration problem with

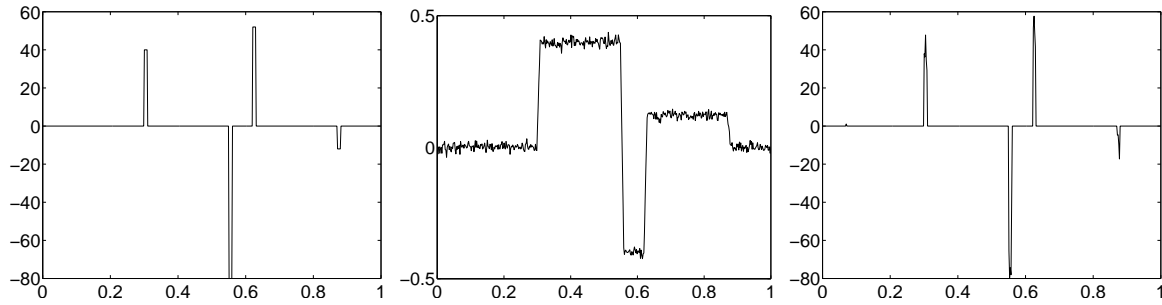


Figure 1. Illustration of data and results of the inverse integration problem. Left: the true solution \bar{u} with $N = 500$, middle: the noisy data f with 5% noise, right: the reconstruction by ℓ^1 minimization with $w_k = 3 \cdot 10^{-3}$ and $\gamma = 5 \cdot 10^5$. The solution was obtained with the semismooth Newton method after 11 iterations with a residual norm of $1.7 \cdot 10^{-10}$.

n	$\Psi(u^n)$	$\ r^n\ $
1	1.3249e+01	6.9764e+05
2	1.0461e+01	2.1698e+02
3	5.3849e+00	2.9586e+02
4	4.8393e+00	8.3922e+02
5	4.2488e+00	1.9864e+02
6	3.0433e+00	1.5474e+02
7	2.8758e+00	3.9127e+01
8	2.8237e+00	3.5658e+01
9	2.7365e+00	2.9485e+01
10	2.5984e+00	7.7932e+00
11	2.5518e+00	1.7423e-10

Table 1. Illustration of the performance of the SSN method for the inverse integration problem. The second column shows the decay of the function value Ψ while the third column shows the norm of the residual. The data is the same as in Figure 1.

problem size N between 100 and 5000. We kept all parameters, as well as the data and the noise level fixed and only refined the discretization of the problem. We stopped the algorithms when a required residual tolerance was reached. Moreover, we checked if the reached functional value was equal for the different methods since the algorithms used different stopping criteria. Table 4 reports CPU times required for the SSN method, for GPSR, `11_1s` and for the iterative thresholding. In Figure 2 the same data is coded graphically. When assuming that the computational cost is $\mathcal{O}(N^\beta)$ we found $\beta = 2.71$ for GPSR, $\beta = 2.70$ for `11_1s`, $\beta = 2.15$ for iterative thresholding, and $\beta = 2.20$ for SSN. Moreover, the constant hidden in the \mathcal{O} notation is considerably smaller for the SSN method. The observed scaling differs from the results reported in [14] which may be due to the different structure of the examples. In [14] the example used a matrix which had

$\ \delta\ $	#iter	CPU-Time (sec)	$\ r\ $
1.0e+00	5	6.66e-01	6.18e-10
1.0e-01	8	7.63e-01	2.19e-09
1.0e-02	12	3.98e-01	7.85e-10
1.0e-03	10	3.11e-01	1.29e-09
1.0e-04	11	3.22e-01	9.85e-10
1.0e-05	11	3.18e-01	2.49e-09

Table 2. Behavior of the SSN method for different noise levels with fixed $w_k = 10^{-5}$. The problem under consideration is the inverse integration, the problem size is $N = 500$ with $\gamma = 5 \cdot 10^5$ throughout. The rightmost column shows the residual norm at convergence.

$\ \delta\ $	#iter	CPU-Time (sec)	$\ r\ $
1.0e-01	9	1.25e-01	6.94e-12
1.0e-02	11	2.61e-01	2.29e-11
1.0e-03	11	2.96e-01	1.47e-10
1.0e-04	11	3.10e-01	2.91e-11
1.0e-05	11	3.25e-01	7.63e-11
1.0e-06	12	3.50e-01	4.16e-11

Table 3. Behavior of the SSN method for different noise levels. The problem under consideration is the inverse integration, the problem size is $N = 500$. We chose $w_k = \|\delta\|$ and $\gamma = 5 \cdot 10^5$ throughout. The rightmost column shows the residual norm at convergence.

all singular values either close to one or zero, while in our example the singular values converge to zero. Hence, it is expected that the empirical scaling of the computational costs differs from problem to problem.

4.2. Deblurring in a Haar basis. As a second example of an ill-posed problem we consider a blurring operator $A : L^2([0, 1]) \rightarrow L^2([0, 1])$ given by $Au = k * u$ with the kernel $k(x) = c(1 + x^2/\lambda^2)^{-1}$ with $\lambda = 0.01$. We choose c such that $\int k dx = 1$ and consider u to be extended periodically to \mathbb{R} in order to evaluate the convolution integral.

In this example we work with a synthesis operator $B : \ell^2 \rightarrow L^2([0, 1])$ mapping coefficients (c_k) to a function $u = \sum_k c_k \psi_k$ where the (ψ_k) form the orthonormal Haar wavelet basis [19]. Hence, the operator under consideration $K = AB$ is a blurring after a Haar wavelet synthesis, see [6, 10] for discussions of ℓ^1 penalty terms in combination with wavelet expansions.

We start with a given function u which is piecewise constant. The data f is computed as $f = Au + \text{noise}$ such that we have 25% relative error, i.e. $\|f - Au\|/\|f\| =$

N	SSN	GPRS	l1_ls	iterthresh
100	3.06e-02	8.29e-01	1.01e+00	2.04e+01
150	3.57e-02	1.47e+00	1.69e+00	4.03e+01
224	5.31e-02	4.28e+00	4.40e+00	1.06e+02
335	1.70e-01	1.16e+01	1.12e+01	2.43e+02
501	2.83e-01	2.99e+01	2.37e+01	5.78e+02
750	6.84e-01	1.36e+02	7.69e+01	1.26e+03
1122	2.20e+00	2.65e+02	2.42e+02	2.59e+03
1679	6.37e+00	1.22e+03	9.95e+02	7.42e+03
2512	1.87e+01	3.64e+03	3.88e+03	1.95e+04
5000	1.28e+02	2.51e+04	3.56e+04	9.45e+04

Table 4. Comparison of the CPU time in seconds for the different algorithms and different sizes of the problem. The problem under consideration is the inverse integration problem with 5% noise and regularization parameter $w_k = 3 \cdot 10^{-3}$.

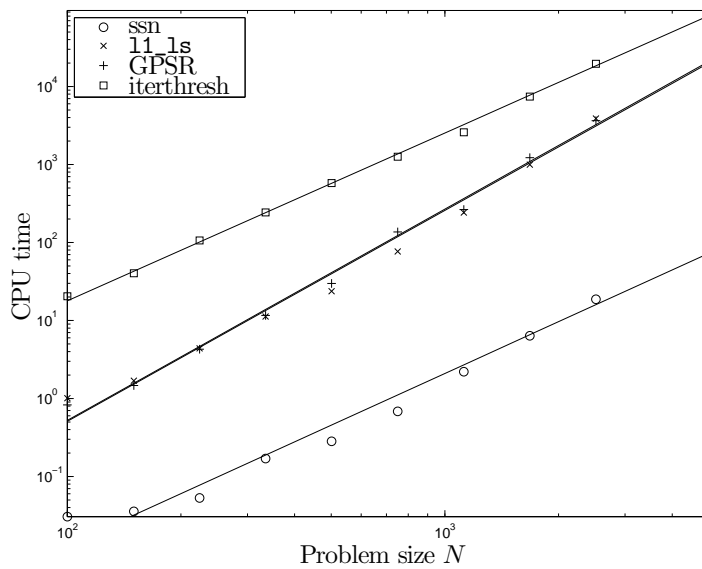


Figure 2. The empirical growth of the computational cost for the different algorithms.

0.25. The Haar coefficients of u have been reconstructed by minimizing (1). As an illustration of ℓ^1 penalties in contrast to classical ℓ^2 regularization we also show the results of the minimization of

$$\frac{1}{2} \|Kc - f\|^2 + \sum_{k=1}^{\infty} w_k |c_k|^2 .$$

Figure 3 and Table 5 show the results of both ℓ^1 and the above ℓ^2 regularization where we discretized the problem to 1024 Haar wavelets. The parameters w_k are independent of k and have been tuned by hand to produce optimal results. Since the original data

is quite sparse in the Haar wavelet basis, the ℓ^1 -reconstruction leads to much better results, as expected from the model. It also turned out that the algorithm is robust with respect to different initial values u^0 . We tested several initial values (starting at zero, at K^*f or at a random position) and the observed convergence behavior was very similar in all cases.

The SSN method converged in six iterations and in 0.3 seconds (for comparison: the GPSR method takes 0.5 seconds and `l1_ls` converged in 5.3 seconds).

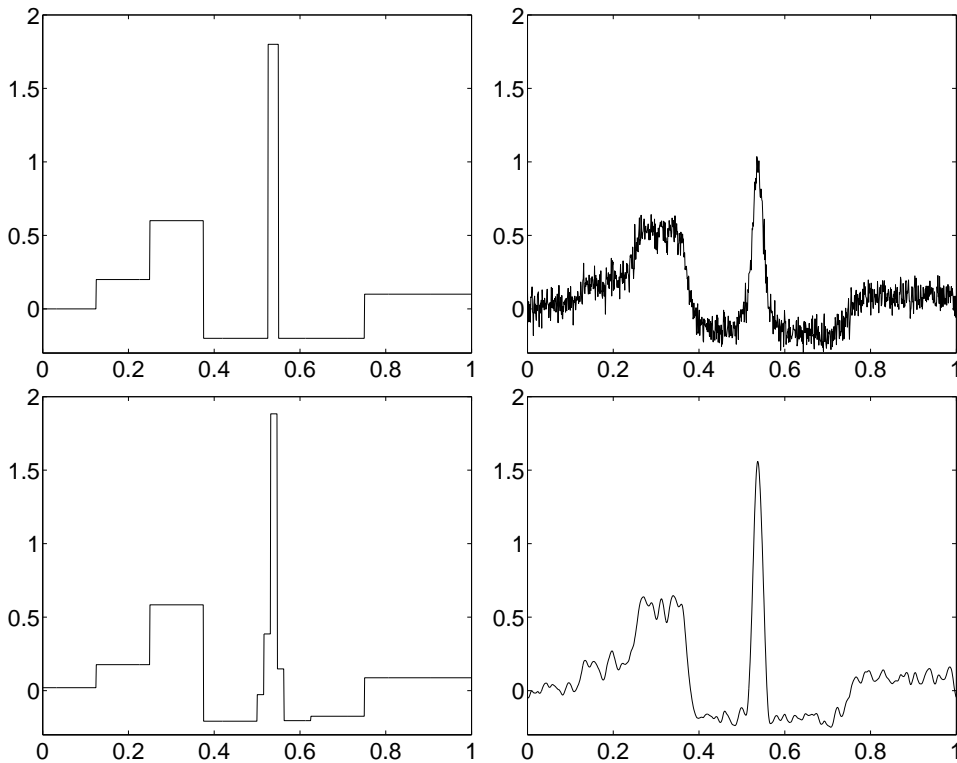


Figure 3. The results of ℓ^1 and ℓ^2 (classical Tikhonov) regularization of deblurring in a Haar basis. Upper left: the true solution u , upper right: the given data f , lower left: the reconstruction by ℓ^1 minimization with $w_k = 0.12$ and $\gamma = 5 \cdot 10^6$, lower right: the reconstruction by ℓ^2 minimization with $w_k = 0.05$.

4.3. Compressive Sampling. In our last example we illustrate the applicability of the SSN method to the decoding problem in compressive sampling alias compressed sensing (CS). In CS one aims at reconstructing a signal from very few linear measurements, see [4, 11] for an introduction to CS. A popular way of decoding a signal from data f which was measured by the observation operator K is to minimize a functional of type (1), see [5]. Our example on compressive sampling is taken from [14]. We obtain an observation operator $K \in \mathbb{R}^{K \times N}$ by first filling it with independent samples of a standard Gaussian distribution and then orthonormalizing the rows. Hence, the operator is not injective but it possesses the so-called restricted isometry property (see [1]) which means that all submatrices consisting of a small number of columns have

n	$\Psi(u^n)$	$\ r^n\ $
1	3.3920e+001	2.9676e+006
2	1.3905e+002	3.2499e+004
3	1.3326e+001	6.2647e+005
4	7.9347e+000	1.7517e+004
5	6.0006e+000	8.0510e-002
6	5.9823e+000	1.5424e-009

Table 5. Illustration of the performance of the SSN method for deblurring in a Haar basis. The second column shows the decay of the function value Ψ while the third column shows the norm of the residual. The data is the same as in Figure 3.

singular values close to one. Especially, submatrices made of a small number of columns are injective. Hence, the SSN method works as long as the active sets are small enough.

In this example we chose $N = 8192$, $K = 512$, and the signal u contained 64 randomly placed ± 1 spikes. The observation f was generated by $f = Ku + \text{noise}$ such that we have 5% relative error. The minimization of (1) with $w = 0.05$ was done with the SSN method with parameter $\gamma = 5 \cdot 10^4$. The SSN method converged in approximately 1.2 seconds in six iterations and the active sets stayed very small during the iteration, see Figure 4 and Table 6. Hence, the SSN method is a promising candidate for the decoding problem in CS.

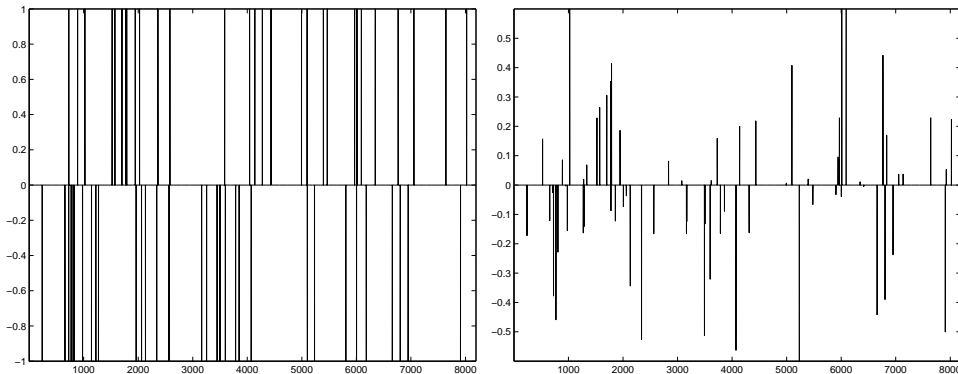


Figure 4. Illustration of data and results of the CS example. Left: the original signal u with $n = 8192$, right: the reconstruction by ℓ^1 minimization with $w_k = 0.05$ and $\gamma = 5 \cdot 10^4$. The solution was obtained with the semismooth Newton method after 6 iterations with a residual norm of approximately $1 \cdot 10^{-11}$.

5 Conclusion

We have shown that the semismooth Newton method applied to Tikhonov functionals with sparsity constraints is a fast algorithm which is easy to implement as an active

n	$\Psi(u^n)$	$\ r^n\ $	$ \mathcal{A} $	$\text{cond}(\mathcal{M}_{\mathcal{A}\mathcal{A}})$
1	2.0774e+00	1.2254e+04	252	44.52
2	7.1752e+00	2.2715e+03	148	12.77
3	2.7379e+00	4.6644e+02	90	6.31
4	1.9997e+00	1.4674e+02	67	4.75
5	1.8386e+00	3.9728e+01	67	4.60
6	1.8361e+00	9.9652e-12	67	

Table 6. Illustration of the performance of the SSN method for CS. The second column shows the decay of the function value Ψ while the third column shows the norm of the residual. The fourth and fifth column show the size of the active set and the condition of the matrix $\mathcal{M}_{\mathcal{A}\mathcal{A}}$ which has to be inverted in the Newton step. The data is the same as in Figure 4.

set method. Each iteration involves the solution of a system of linear equations on the active coefficients only. Our numerical experiments show that these systems stay reasonably small during the iteration and are also very well conditioned. In addition, the experiments indicate that the SSN method compares favorably with existing state-of-the-art methods when applied to ill-posed problems. While we investigated only the local convergence behavior, the numerical experiments indicate that our method is robust with respect to the initial value of the iteration. However, the convergence is slow as long as the iterates are far from the minimizer and it gets faster when the solution is approached. The global convergence properties are not yet explained by our theory and need further investigation. Another direction for further research is globalization of the method e.g., by the use of an appropriate merit function, and line search or trust region methods.

Appendix

We define for $u \in \ell^2$ and $h \in \mathcal{H}$

$$F(u) = \sum_{k=1}^{\infty} w_k |u_k|, \quad G(h) = \frac{1}{2} \|h - f\|_{\mathcal{H}}^2$$

and calculate their conjugate (polar) functions, see [12, Ch. I.4]. We have

$$\begin{aligned} F^*(p) &= \sup_{u \in \ell^2} (\langle p, u \rangle - F(u)) = \sup_u (\langle p, u \rangle - \sum_{k=1}^{\infty} w_k |u_k|) \\ &= \sup_u \left(\sum_{k=1}^{\infty} (p_k - w_k \text{sign } u_k) u_k \right) = \begin{cases} 0, & \text{if } |p_k| \leq w_k \text{ for all } k \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

For G , we obtain

$$G^*(p) = \sup_{h \in \mathcal{H}} (\langle p, h \rangle - G(h)) = \sup_h (\langle p, h \rangle - \frac{1}{2} \|h - f\|_{\mathcal{H}}^2) = \frac{1}{2} \|p\|_{\mathcal{H}}^2 + \langle p, f \rangle,$$

since the supremum is attained at $h = p + f$.

References

- [1] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. To appear in *Constructive Approximation*, 2008.
- [2] Kristian Bredies and Dirk A. Lorenz. Iterative soft-thresholding converges linearly. Submitted for publication, arXiv.org/abs/0709.1598., 2007.
- [3] Kristian Bredies and Dirk A. Lorenz. Iterated hard shrinkage for minimization problems with sparsity constraints. *SIAM Journal on Scientific Computing*, 30(2):657–683, 2008.
- [4] Emmanuel J. Candès. Compressive sampling. In *Proc. International Congress of Mathematics*, pages 1433–1452, 2006.
- [5] Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transaction on Information Theory*, 51(12):4203–4215, 2005.
- [6] Antonin Chambolle, Ronald A. DeVore, Namyong Lee, and Bradley J. Lucier. Nonlinear wavelet image processing: Variational problems, compression and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7:319–335, 1998.
- [7] Xiaojun Chen. Superlinear convergence and smoothing quasi-Newton methods for nonsmooth equations. *Journal of Computational and Applied Mathematics*, 80(1):105–126, 1997.
- [8] Xiaojun Chen, Zuhair Nashed, and Liqun Qi. Smoothing methods and semismooth methods for nondifferentiable operator equations. *SIAM Journal on Numerical Analysis*, 38(4):1200–1216, 2000.
- [9] Patrick L. Combettes and Valérie R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4(4):1168–1200, 2005.
- [10] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications in Pure and Applied Mathematics*, 57(11):1413–1457, 2004.
- [11] David Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [12] Ivar Ekeland and Roger Temam. *Convex Analysis and Variational Problems*. North-Holland, Amsterdam, 1976.
- [13] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 2000.
- [14] Mário A. T. Figueiredo, Robert D. Nowak, and Stephen J. Wright. Gradient projection for sparse reconstruction: Applications to compressed sensing and other inverse problems. To appear in *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 2008.
- [15] Martin Hanke and Otmar Scherzer. Inverse problems light: Numerical differentiation. *The American Mathematical Monthly*, 108(6):512–521, 2001.
- [16] Michael Hintermüller, Kazufumi Ito, and Karl Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.
- [17] Seung-Jean Kim, Kwangmoo Koh, Michael Lustig, Stephen Boyd, and Dimitry Gorinevsky. A method for large-scale ℓ_1 -regularized least squares problems with applications in signal processing and statistics. To appear in *IEEE Journal on Selected Topics in Signal Processing*, 2008.

- [18] Dirk A. Lorenz. Convergence rates and source conditions for Tikhonov regularization with sparsity constraints. Submitted for publication, [arXiv.org/abs/0801.1774](https://arxiv.org/abs/0801.1774), 2008.
- [19] Alfred Karl Louis, Peter Maaß, and Andreas Rieder. *Wavelets: Theory and Application*. Wiley, Chichester, 1997.
- [20] Ronny Ramlau and Gerd Teschke. A Tikhonov-based projection iteration for nonlinear ill-posed problems with sparsity constraints. *Numerische Mathematik*, 104(2):177–203, 2006.
- [21] Elena Resmerita. Regularization of ill-posed problems in Banach spaces: convergence rates. *Inverse Problems*, 21(4):1303–1314, 2005.
- [22] Arnd Rösch and Karl Kunisch. A primal-dual active set strategy for a general class of constrained optimal control problems. *SIAM Journal on Optimization*, 13(2):321–334, 2002.
- [23] Frank Schöpfer, Alfred K. Louis, and Thomas Schuster. Nonlinear iterative methods for linear ill-posed problems in Banach spaces. *Inverse Problems*, 22:311–329, 2006.
- [24] Georg Stadler. Elliptic optimal control problems with l^1 -control cost and applications for the placement of control devices. To appear in *Computational Optimization and Applications*, 2008.
- [25] Michael Ulbrich. Semismooth Newton methods for operator equations in function spaces. *SIAM Journal on Control and Optimization*, 13(3):805–842, 2003.
- [26] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications II/B: Nonlinear Monotone Operators*. Springer, 1990.