

AN ADAPTIVE INVERSE SCALE SPACE METHOD FOR COMPRESSED SENSING

MARTIN BURGER, MICHAEL MÖLLER, MARTIN BENNING, AND STANLEY OSHER

ABSTRACT. In this paper we introduce a novel adaptive approach for solving ℓ^1 -minimization problems as frequently arising in compressed sensing, which is based on the recently introduced inverse scale space method. The scheme allows to efficiently compute minimizers by solving a sequence of low-dimensional nonnegative least-squares problems.

We provide a detailed convergence analysis in a general setup as well as refined results under special conditions. In addition, we discuss experimental observations in several numerical examples.

1. INTRODUCTION

Compressed sensing and techniques exploiting sparsity in data analysis, image processing, and inverse problems recently gained enormous interest. Representing unknowns for (underdetermined) systems of linear equations in appropriate bases or dictionaries can be reformulated as finding the sparsest solution of a linear system

$$(1.1) \quad Au = f,$$

with a matrix $A \in \mathbb{R}^{m \times n}$, usually with m much smaller than n . In a natural way sparsity is measured via the number of nonzero elements in the vector $u \in \mathbb{R}^n$ (the ℓ^0 -norm), which however leads to highly nonconvex problems when trying to find the sparsest solution. In order to avoid the high complexity of ℓ^0 -minimization, one uses a convex relaxation to minimize the ℓ^1 -norm instead, i.e.,

$$(1.2) \quad \|u\|_1 \rightarrow \min_u \quad \text{subject to } Au = f.$$

Various important results have been obtained on the equivalence of ℓ^0 - and ℓ^1 -minimization under different conditions; we refer e.g. to [10, 11, 17, 16].

Instead of a nonconvex problem, the ℓ^1 approach solves a convex problem with linear equality constraints, however, still with a nonsmooth objective functional. Thus, the efficient numerical solution is often a challenge, in particular, in large dimensions. A similar argument applies to regularized problems used in the case of noisy data, where one minimizes

$$(1.3) \quad E_\alpha(u) = \frac{1}{2} \|Au - f\|^2 + \alpha \|u\|_1.$$

Received by the editor February 23, 2011 and, in revised form, July 11, 2011.

2010 *Mathematics Subject Classification*. Primary 49M29, 90C25, 65F20, 65F22.

Key words and phrases. Compressed sensing, inverse scale space, sparsity, adaptivity, greedy methods.

The work of MB and MB has been supported by the German Research Foundation DFG through the project *Regularization with Singular Energies*. M.M. and S.O. were supported by NSF grants DMS-0835863, DMS-0914561, DMS-0914856 and ONR grant N00014-08-1119. M.M. also acknowledges the support of the German Academic Exchange Service (DAAD).

Methods proposed for these problems are, e.g., iterative thresholding [2, 15] and variants [18, 24, 30]. A method receiving particular attention due to its efficiency for many problems is the Bregman iteration [25, 33], respectively, its linearized version [9, 8]. These methods are formulated in general for problems of the form

$$(1.4) \quad J(u) \rightarrow \min_{u \in X} \quad \text{subject to } Au = f,$$

where $J : X \rightarrow \mathbb{R} \cup \{\infty\}$ is a convex functional on a Banach space X and $A : X \rightarrow Y$ is a bounded linear operator between Banach spaces. The Bregman iteration constructs a sequence u_k as minimizers of

$$(1.5) \quad E_k(u) = \frac{\lambda}{2} \|Au - f\|^2 + J(u) - \langle p_{k-1}, u \rangle,$$

where p_{k-1} is an element of the subdifferential of J at u_{k-1} , $p_{k-1} \in \partial J(u_{k-1}) = \{p : J(u) - J(u_{k-1}) - \langle p, u - u_{k-1} \rangle \geq 0 \forall u\}$. Each step is a penalized least squares problem with the (generalized) *Bregman distance* between u and u_{k-1} being the regularization functional. From the optimality condition a simple update scheme for the subgradients can be obtained as

$$(1.6) \quad p_k = p_{k-1} + \lambda A^*(f - Au_k),$$

which also allows to prove equivalence of the Bregman iteration to the well-known augmented Lagrangian method. The linearized Bregman iteration uses a first-order Taylor expansion of the least-squares term around the last iterate, i.e., u_k is determined from minimizing

$$(1.7) \quad E_k(u) = \frac{\lambda}{2} \|Au_{k-1} - f\|^2 + \lambda \langle A(u - u_{k-1}), Au_{k-1} - f \rangle + J(u) - \langle p_{k-1}, u \rangle.$$

The dual variable is obtained correspondingly from

$$(1.8) \quad p_k = p_{k-1} + \lambda A^*(f - Au_{k-1}).$$

The application to compressed sensing appears obvious for $J(u) = \|u\|_1$, care needs to be taken, however, for the linearized Bregman iterations since the functional E_k is possibly unbounded from below in the ℓ^1 -case, one therefore frequently uses the elastic net

$$(1.9) \quad J(u) = \frac{1}{2\delta} \|u\|_2^2 + \|u\|_1$$

with δ large. This does not harm the convergence to an ℓ^1 -minimizing problem, equivalence can be shown for δ above a threshold (cf. [32]).

Interpreting λ as a time step of an evolution, one observes that the Bregman and linearized Bregman methods are backward (respectively forward) time stepping on the evolution equation (respectively inclusion)

$$(1.10) \quad \partial_t p(t) = A^*(f - Au(t)), \quad p(t) \in \partial J(u(t)),$$

which is called *inverse scale space method* and analyzed in [5]. The inverse scale space method appears not to be practical as a minimization technique at first glance, since it is a nonlinear evolution equation. However, for total variation regularization it has been observed already in numerical experiments that the evolution has a discrete nature (also confirmed partly by theoretical results; cf. [5, 4, 3]). There seem to be only a few time steps where the solution u actually changes, while the dual variable evolves continuously. This observation is one of the motivations for this paper, in which we aim to obtain a similar or even more precise characterization

of the inverse scale space for the compressed sensing setup (1.2) in detail and deduce an efficient numerical scheme.

Our second motivation are adaptive discretizations of operator equations (cf. [12, 13, 14]), in particular, greedy techniques, which are iterative methods that slowly but efficiently increase the number of degrees of freedom (often by one in each iteration step). It is rather natural to look for such adaptive discretizations also in the compressed sensing framework, because one looks for solutions with only few degrees of freedom (associated with the nonzero entries in u). One might hope to construct methods that slowly increase the support of the iterates towards the one of the sparsest solution, solving only (low-dimensional) linear problems on the support of the iterates in each step. Usually such adaptive methods are guided by a posteriori error analysis and as we shall see below, a simple a posteriori approach to ℓ^1 -minimization indicates that the supremum of $A^*(Au - f)$ should be made small, which is exactly what the inverse scale space achieves (examine for instance the right-hand side in (1.10) and the convergence analysis in [5, 4] reviewed in the next section).

Our motivations suggest that the inverse scale space method applied to the compressed sensing setup will provide an efficient and adaptive approach when carefully analyzed and implemented, and this is indeed one of our main findings in this paper. We will show that there exists a discrete set of time steps t_k at which the primal solution of the inverse scale space is changing and where $u_k = u(t_k)$ can be computed by solving low-dimensional systems of linear equations with nonnegativity constraints (which can often be even omitted in practice). The evolution of the dual variable is characterized as linear between the discrete time steps and hence can be computed explicitly as well. Moreover, the discrete time steps can be obtained from the supremum norm of the dual variable (respectively the residuum). Finally, the behaviour of p also provides the key to the adaptive refinement, i.e., the small index sets on which A has to be considered and the linear equations need to be solved. In addition to the setup in (1.2) we shall also consider the regularized problem of minimizing the functional in (1.3). This yields an inverse scale space flow of the form

$$(1.11) \quad \partial_t p(t) = -\alpha p(t) + A^*(f - Au(t)), \quad p(t) \in \partial J(u(t)),$$

which has not yet been investigated in detail in literature, but for which we can obtain analogous results to the case $\alpha = 0$.

The convergence behavior of the explicit discretization of the inverse scale space flow, also known as linearized Bregman, has been studied before by Osher et al. in [26] where it was discovered that the primal variable alternates between quickly converging to a new solution and stagnating, i.e., staying constant for long times. This fact was used to determine the stagnation time and immediately “kick” the primal variable to a time where the next change of solution happens. Hence, “kicking” is the first work to use the discrete nature of the inverse scale space flow. However, opposed to the method we propose in this paper, linearized Bregman is a discretization of (1.10). By analyzing the behavior of the subgradient we will be able to solve the (continuous) flow (1.10) exactly, without discretization of the underlying differential equation. The times at which the solution changes can be calculated exactly such that a stagnation is entirely avoided. Any change in the continuous inverse scale space flow solution happens instantaneously at discrete times.

Besides linearized Bregman with kicking, our approach is also related to the technique of orthogonal matching pursuit (OMP) (cf. [27, 20, 30]), which is a greedy method for solving the sparse approximation problem and iteratively adds components to the support of u whose correlation to the current residual is maximal. To be more precise the pseudo code for OMP is given in Algorithm 1 below. Although the motivation of OMP is very different than the one of inverse scale space methods, we will see that the first iteration of OMP coincides with the first step of our proposed approach to solve (1.10). We will discuss the differences to OMP in more detail throughout the paper and will furthermore provide a numerical comparison in Section 5.

Algorithm 1. Orthogonal Matching Pursuit.

```

1. Parameters:  $A$ ,  $f$ , threshold  $> 0$ 
2. Initialization:  $r_0 = f$ ,  $I_0 = \emptyset$ 
while  $\|r_k\| > \text{threshold}$  do
    Compute  $I_k = I_{k-1} \cup i$  with  $i$  such that  $|(A^T r_k)_i| = \|A^T r_k\|_\infty$ 
    Compute  $u_k = \arg \min_u \{\|AP_{I_k} u - f\|^2\}$  and set  $(u_k)_i = 0$  for  $i \notin I_k$ 
    Update  $r_{k+1} = f - AP_{I_k} u_k$ 
end while
return  $u_k$ 

```

Another technique that aims at iteratively reconstructing the support of a sparse signal is iterative support detection (ISD) proposed by Wang and Yin in [31]. They iteratively solve a truncated ℓ^1 -minimization problem on the coefficients that are currently not in the detected support and update the set of support coefficients by thresholding. Similar to the framework we propose in this paper, ISD can also change the support of the solution arbitrarily and does not lead to increasing, nested supports as OMP. However, ISD is not based on the inverse scale space flow (1.10) and does not aim at solving (1.2), which is a fundamental difference to our method. In fact, the first iteration of ISD requires to fully solve (1.2) already, such that the method proposed in this paper could even be combined with ISD.

The remaining part of the paper is organized as follows: In Section 2 we provide a review of the most important properties and convergence analysis of inverse scale space methods as well as their extension to the case of regularized variational problems, which allows for a refined convergence analysis. Section 3 discusses the specialization of inverse scale space methods to the compressed sensing setup and provides a characterization allowing to adaptively choose time steps and low-dimensional discretizations in each step. In Section 4 we provide some further convergence analysis and subsequently discuss various numerical test examples illustrating the behaviour of the methods and a comparison to the well-known OMP method in Section 5.

2. INVERSE SCALE SPACE METHODS

In the following we provide some basic facts about Bregman iterations and inverse scale space methods arising as their asymptotic limit. In this section we consider the general case as described in (1.4). In Section 2.2 we shall further discuss the

application to the regularized problem

$$(2.1) \quad \frac{1}{2} \|Au - f\|^2 + \alpha J(u) \rightarrow \min_{u \in X}.$$

The starting point of the analysis is the Bregman distance associated with the functional J . For a subgradient $p \in \partial J(u)$ the Bregman distance is defined as

$$(2.2) \quad D_J^p(v, u) = J(v) - J(u) - \langle p, v - u \rangle.$$

Using the Bregman distance between subsequent iterations as a penalty in the least-squares problem, we obtain the Bregman iteration (cf. [25]), computing u_k as the solution of

$$(2.3) \quad \frac{\lambda}{2} \|Au - f\|^2 + D_J^{p_{k-1}}(u, u_{k-1}) \rightarrow \min_{u \in X}.$$

Note that since by subtracting constant terms does not change the minimizer, this is equivalent to (1.5).

The optimality condition characterizing the minimizer u_k is given by

$$(2.4) \quad \lambda A^*(Au_k - f) + p_k - p_{k-1} = 0.$$

Interpreting $\Delta t = \lambda$ as a time step and $p_k = p(k\Delta t)$ we find the equivalent relation

$$\frac{p(t) - p(t - \Delta t)}{\Delta t} = A^*(f - Au(t)),$$

which is a backward Euler discretization of the time-continuous flow (1.10), i.e.,

$$\partial_t p(t) = A^*(f - Au(t)), \quad p(t) \in \partial J(u(t)),$$

called inverse scale space method (ISS). The inverse scale space flow is a differential inclusion, it can also be formulated as a dual gradient flow using the relation $p = A^*q$ for some q , which allows us to write

$$(2.5) \quad \partial_t q(t) = f - Au(t), \quad u(t) \in \partial_p J^*(A^*q),$$

hence

$$(2.6) \quad \partial_t q(t) \in -\partial E^*(q),$$

with the dual energy functional

$$(2.7) \quad E^*(q) = J^*(A^*q) - \langle f, q \rangle,$$

where J^* denotes the convex conjugate of J (cf. [19]).

2.1. Convergence properties of the inverse scale space flow (3.1). As a consequence of the limit from the Bregman iteration and the obvious decrease of the least-squares functional in the Bregman iteration (cf. [25]) we find

$$(2.8) \quad \|Au(t) - f\| \leq \|Au(s) - f\|, \quad \forall t \geq s.$$

A second useful property concerns the decrease of the Bregman distance and the dissipation of the least-squares functional. If \hat{u} is a solution of $Au = f$ minimizing J then taking the duality product with $u - \hat{u}$ reveals

$$(2.9) \quad \frac{d}{dt} D_J^{p(t)}(\hat{u}, u(t)) = -\|Au(t) - f\|^2.$$

From this inequality one can infer the major convergence properties of the inverse scale space method, namely

$$(2.10) \quad \|Au(t) - f\| = \mathcal{O}(t^{-1/2})$$

and, under appropriate conditions on J , the weak or weak-* convergence of $u(t)$ to solutions of $Au = f$ with minimal J along subsequences. Improved convergence properties can be obtained for data satisfying a *source condition*

$$(2.11) \quad \exists \hat{q} : \quad A^* \hat{q} \in \partial J(\hat{u}).$$

Under such a condition one further obtains (cf. [7])

$$(2.12) \quad D_J^{p(t)}(\hat{u}, u(t)) = \mathcal{O}(t^{-1/2})$$

and $q(t)$ remains bounded.

In the compressed sensing setup this means that $u(t)$ converges to an ℓ^1 -minimizing solution of $Au = f$ (along subsequences if the latter is not unique). Moreover, (2.11) is automatically satisfied, which can be seen from the optimality condition in finite dimensions (which is the Lagrange-multiplier corresponding to $Au = f$). Thus, the Bregman distance with respect to the ℓ^1 -norm converges at least at order $t^{-1/2}$.

In addition to the above properties some regularizing features of the inverse scale space method have been shown under appropriate stopping rules for noisy data f ; we refer to [5] for further discussion.

2.2. Inverse scale space methods for regularized problems. In the case of regularized problems one aims at minimizing the functional

$$(2.13) \quad E(u) = \frac{1}{2} \|Au - f\|^2 + \alpha J(u),$$

and corresponding Bregman and inverse scale space methods can be constructed as well by just replacing the least-squares functional with E . This finally leads to the flow (1.11), i.e.,

$$\partial_t p(t) = A^*(f - Au(t)) - \alpha p(t), \quad p(t) \in \partial J(u(t)),$$

for which analogous reasoning as in the case of the unregularized problem can be carried out, respectively some results can even be improved due to the presence of p on the right-hand side. First of all, one obtains a decrease of the objective functional, i.e.,

$$(2.14) \quad E(u(t)) \leq E(u(s)), \quad \forall t \geq s.$$

Concerning the decrease of the Bregman distance we can show a stronger result:

Proposition 1. *Let (u, p) be a solution of (1.11) for $\alpha > 0$. Then, for \hat{u}_α being a minimizer of E , the estimate*

$$(2.15) \quad D_J^{p(t)}(\hat{u}_\alpha, u(t)) \leq e^{-\alpha t} J(\hat{u}_\alpha)$$

holds.

Proof. Taking the duality product of (1.11) with $u - \hat{u}_\alpha$ yields

$$\begin{aligned} \frac{d}{dt} D_J^p(\hat{u}_\alpha, u) &= -\langle A^*(Au - A\hat{u}_\alpha + A\hat{u}_\alpha - f), u - \hat{u}_\alpha \rangle - \alpha \langle p, u - \hat{u}_\alpha \rangle \\ &= -\|Au - A\hat{u}_\alpha\|^2 - \alpha \langle p - \hat{p}_\alpha, u - \hat{u}_\alpha \rangle \\ &\leq -\alpha D_J^p(\hat{u}_\alpha, u), \end{aligned}$$

where we have inserted the optimality condition for \hat{u}_α with subgradient $\hat{p}_\alpha \in \partial J(\hat{u}_\alpha)$ in the second line. The Gronwall inequality finally yields the assertion. \square

A similar result can be shown for the dual variable q satisfying $p = A^*q$. Using

$$\hat{q}_\alpha = \frac{1}{\alpha}(f - A\hat{u}_\alpha)$$

one can show

$$\|q(t) - \hat{q}_\alpha\| \leq e^{-\alpha t}\|\hat{q}_\alpha\|,$$

which by the continuity of A^* also implies the exponential convergence of $p = A^*q$.

3. INVERSE SCALE SPACE METHODS FOR COMPRESSED SENSING

3.1. The aISS flow for (1.2). In the following we are going to investigate the behaviour of inverse scale space methods in the compressed sensing setup

$$(3.1) \quad \partial_t p(t) = A^T(f - Au(t)), \quad p(t) \in \partial\|u\|_1.$$

Notice that the subdifferential of the ℓ^1 -norm can be characterized componentwise by

$$(3.2) \quad p \in \partial\|u\|_1 \Leftrightarrow \begin{cases} p_i = \text{sign}(u_i), & \text{if } u_i \neq 0, \\ |p_i| \leq 1, & \text{else.} \end{cases}$$

We start with a simple result that can be shown in a very general way for one-homogeneous regularization functionals (using a dual norm of the initial residual, cf. [21, 6]):

Lemma 1. *For*

$$(3.3) \quad t < t_1 := \frac{1}{\|A^T f\|_\infty}$$

a solution (u, p) of the inverse scale space flow (3.1) is given by

$$(3.4) \quad u(t) = 0, \quad p(t) = tA^T f.$$

Proof. We immediately see

$$\partial_t p(t) = A^T f = A^T(f - Au(t))$$

and

$$\|p(t)\|_\infty = t\|A^T f\|_\infty < t_1\|A^T f\|_\infty = 1,$$

thus

$$p(t) \in \partial\|0\|_1 = \partial\|u(t)\|_1. \quad \square$$

The observation in Lemma 1 is the basis for further characterizing the inverse scale space flow for larger times. We expect changes in the primal variable u only to occur at some discrete time steps, when some $|p_i(t)|$ reaches the value one and that p behaves linearly in the intermediate times. It remains to characterize u at the discrete time steps t_k . This can be understood from the limit of the Bregman iteration

$$(3.5) \quad \frac{1}{2}\|Au - f\|^2 + \alpha(\|u\|_1 - p \cdot u)$$

as $\alpha \rightarrow \infty$. In order to obtain a minimum we expect the Bregman distance to go to zero, i.e., $p \cdot u = \|u\|_1$ (thus $p \in \partial\|u\|_1$), and the squared norm to be minimized subject to this constraint. In this way we can indeed compute the detailed behaviour of the inverse scale space flow:

Theorem 1. *There exists a sequence of times*

$$0 = t_0 < t_1 < t_2 < \dots$$

such that

$$(3.6) \quad u(t) = u(t_k), \quad p(t) = p(t_k) + (t - t_k)A^T(f - Au(t_k))$$

for $t \in [t_k, t_{k+1})$ is a solution of the inverse scale space flow (3.1), where $u(t_k)$ is a solution of

$$(3.7) \quad \|Au - f\| \rightarrow \min_{u, p(t_k) \in \partial \|u\|_1}.$$

Moreover, $t_{k+1} = \infty$ if and only if $A^T Au(t_k) = A^T f$.

Proof. Due to Lemma 1 the assertion clearly holds for $k = 0$ (with $t_0 = 0$), noticing that

$$p(0) = 0 \in \partial \|u(0)\|_1 = \partial \|0\|_1.$$

Now we proceed inductively. Given $u(t_k)$ and $p(t_k)$ we compute

$$(3.8) \quad t_{k+1} = \min\{t \mid t > t_k, \exists j : |p_j(t)| = 1, u_j(t_k) = 0, p_j(t) \neq p_j(t_k)\},$$

where

$$(3.9) \quad p_j(t) = p_j(t_k) + (t - t_k)e_j \cdot A^T(f - Au(t_k))$$

holds.

Now $u(t_k)$ minimizes $\|Au - f\|$ subject to the constraint $p(t_k) \in \partial \|u\|_1$, which we can rewrite as a linearly constrained quadratic problem of minimizing $\|Au - f\|^2$ subject to

$$\begin{aligned} u_j &\geq 0 && \text{if } p_j = 1, \\ u_j &\leq 0 && \text{if } p_j = -1, \\ u_j &= 0 && \text{if } -1 < p_j < 1. \end{aligned}$$

From the optimality condition we obtain:

- $u_j(t_k) > 0$ or $u_j(t_k) < 0$ implies $e_j \cdot A^T(Au(t_k) - f) = 0$, hence $p_j(t) = p_j(t_k) = \pm 1$.
- $u_j(t_k) = 0$ and $p_j(t_k) = 1$ implies $e_j \cdot A^T(Au(t_k) - f) \geq 0$, hence $-1 \leq p_j(t) \leq p_j(t_k) = 1$ for t small.
- $u_j(t_k) = 0$ and $p_j(t_k) = -1$ implies $e_j \cdot A^T(Au(t_k) - f) \leq 0$, hence $1 \geq p_j(t) \geq p_j(t_k) = -1$ for t small.
- $u_j(t_k) = 0$ and $p_j(t_k) \in (-1, 1)$ implies $p_j(t) \in [-1, 1]$ for t small.

Hence the construction yields a solution in $[t_k, t_{k+1})$ and t_{k+1} is well defined. The existence of $u(t_{k+1})$ follows from a standard result for quadratic programs. \square

Theorem 1 provides a direct way to formulate the inverse scale space method as an adaptive scheme for compressed sensing. We will refer to this method as the adaptive inverse scale space method (aISS):

Algorithm 2. Adaptive Inverse Scale Space Method

```

1. Parameters:  $A$ ,  $f$ , threshold  $\geq 0$ 
2. Initialization:  $t_1 = 1/\|A^T f\|_\infty$ ,  $p(t_1) = t_1 A^T f$ ,  $I_1 = \{i \mid |p_i(t_1)| = 1\}$ 
while  $\|Au(t_k) - f\| > \text{threshold}$  do
    Compute  $u(t_k) = \arg \min_u \{\|AP_{I_k} u - f\|^2\}$  subject to  $u(t_k)p(t_k) \geq 0$ 
    Obtain  $t_{k+1}$  as the minimal time for which (3.8) holds
    Update the dual variable  $p(t)$  via (3.9) with  $t = t_{k+1}$ 
    Compute  $I_{k+1} = \{i \mid |p_i(t_{k+1})| = 1\}$ 
end while
return  $u(t_k)$ 

```

We want to mention that the solution of the quadratic programming problem in each step is a very low-dimensional one, since we can directly set $u_j(t_k) = 0$ for $|p_j(t_k)| < 1$. Hence we can minimize the problem on the lower-dimensional index set I_k of components where $|p_j(t_k)| = 1$. In most cases we expect the solution to be the same as $(AP_{I_k})^\dagger f$, hence we might try to solve the low-dimensional least-squares problem first and then check the signs of the solution.

3.2. Similarities to other compressed sensing methods.

3.2.1. *OMP*. The resulting characterization of the inverse scale space method is reminiscent to greedy methods for compressed sensing, in particular, orthogonal matching pursuit (see Algorithm 1). We can easily see the differences concerning the structure of the algorithm

- OMP restricts the index set similar to the inverse scale space method, but does not enforce a sign constraint.
- OMP only adds single indices in each iteration, while it is possible to change the index set arbitrarily in inverse scale space methods (practical observations confirm, however, a change of only one index in a vast majority of cases).
- OMP, brought to the inverse scale space notation, uses the supremum norm of $A^T(f - Au(t_k))$ to select relevant indices, while the inverse scale space method uses $sp(t_k) + A^T(f - Au(t_k))$ for varying s (related to $t - t_k$).

We expect the last point to be the major change from OMP to inverse scale space methods, the update of the dual variable encodes some history and improves the convergence behaviour. In situations where OMP performs well, the first two points probably have no major differences, since we expect to automatically find correct signs by solving the unconstrained least-squares problem and we expect the index set to increase by one in each step also in the inverse scale space method. The sign constraint and the different update of the index set (in particular, also the chance to make it smaller) is expected to yield improved behaviour in situations where OMP does not perform well (see also the numerical examples in Section 5), it also reflects the convergence of the inverse scale space method in arbitrary situations.

3.2.2. *Linearized Bregman with kicking*. While the OMP and CoSaMP are greedy methods for solving sparse recovery problems and can give answers very different from the aISS algorithm results, the linearized Bregman method is just a different numerical scheme to determine the solution to (1.2). Although it does not exactly

determine the ℓ^1 -minimizing solution, but rather the solution that minimizes an elastic net of the form (1.9), its minimizer can be shown to coincide with the solution to (1.2) for a large enough δ [32]. As discussed earlier, linearized Bregman solves (1.10) with a forward discretization resulting in the minimization (1.7) and then updates (1.8), while our method determines the solution to (1.10) without any discretization. It is interesting to see that the discrete nature of (1.10) was used in a modification of the linearized Bregman algorithm proposed in [26] to choose adaptive time steps and therefore speed up the algorithm.

It was observed that the iterates of the linearized Bregman algorithm sometimes stagnate and remain constant for many iterations. In some sense, the idea of [26] was to use the discrete form of the subgradient update we are proposing in this paper, i.e., not to simply use the subgradient update (1.8), but to calculate the smallest $t_{k+1} \in \mathbb{N}$ such that the solution changes for a subgradient update of the form

$$(3.10) \quad p^{k+1} = p^k + t_{k+1}A^T(f - Au^k).$$

The full linearized Bregman algorithm with kicking is given as Algorithm 3 below.

Algorithm 3. Linearized Bregman Method with kicking

```

1. Parameters:  $A, f$ , threshold  $\geq 0, \lambda, \delta$ 
2. Initialization:  $u^1 = 0, p^1 = 0$ 
while  $\|Au^k - f\| > \text{threshold}$  do
  if  $u^{k+1} \approx u^k$  then
    Find the smallest  $t^{k+1} \in \mathbb{N}$  such that the support of  $u^{k+1}$  determined by
    (3.11) 
$$p^{k+1} = p^k + t^{k+1}A^T(f - Au^k),$$

    (3.12) 
$$u^{k+1} = \delta \text{shrink}(p^{k+1}, \frac{1}{\lambda})$$

    changes and update accordingly.
  else
    do update with  $t^{k+1} = 1$ .
  end if
end while
return  $u^k$ 

```

The linearized Bregman algorithm with kicking therefore implicitly uses the discrete nature of the inverse scale space flow. However, we should mention that due to the additional ℓ^2 term from the elastic net, the true inverse scale space flow with respect to the elastic net regularization, will not be piecewise constant. It can be seen as a smoother approximation of the true flow. The advantage of the linearized Bregman algorithm is that each step can be calculated explicitly and does not involve the solution of any linear system or minimization problem. However, we expect each iteration of the linearized Bregman algorithm to be less effective than an iteration of the aISS algorithm. In Section 5.2 we show numerical examples for the convergence of the linearized Bregman algorithm with kicking and discuss advantages and disadvantages with respect to aISS.

3.2.3. *Other methods.* Many methods for solving sparse approximation problems have been proposed, including various different minimization algorithms for solving the convex relaxation, i.e., ℓ^1 -minimization, as well as greedy methods that do not have some of the drawbacks that OMP has. For instance, the greedy method CoSaMP [22] can change its support arbitrarily. However, the algorithm we are proposing solves the ℓ^1 -minimization problem (1.2) and a general comparison of greedy algorithms with solutions of (1.2) would be a whole new topic. In this paper we decided to focus on the comparison with OMP, because it is a closely related greedy method, and linearized Bregman iteration with kicking, because it contains similar ideas for solving (1.2).

3.3. **The aISS flow for (1.3).** In analogy to Theorem 1 and Algorithm 2 we can derive similar results for the regularized inverse scale space flow (1.11). Again, one can show that the inverse scale space flow (1.11) stays piecewise constant in time. Thus, in each time interval $[t_k, t_{k+1}[$ in which $u(t) = u(t_k)$, we are left with the simple differential equation

$$(3.13) \quad \partial_t p(t) = -\alpha p(t) + A^T(f - Au(t^k)),$$

which yields a piecewise exponential behavior of the subgradient. In case of $t < t_1$, which implies $u(t) \equiv 0$, the differential equation (3.13) can be solved for p via

$$(3.14) \quad p(t) = \frac{A^T f}{\alpha} (1 - \exp(\alpha t)) .$$

Hence, we can compute

$$p(t_1) = (A^T f) (1 - \exp(\alpha t_1)) / \alpha \quad \text{and} \quad t_1 = -\log (1 - \alpha / \|A^T f\|_\infty) / \alpha,$$

again due to continuous extension. In analogy to the standard aISS the update of the dual variable (3.9) for the regularized aISS can be derived via

$$(3.15) \quad p_j(t) = \left(p_j(t_k) - \frac{1}{\alpha} e_j \cdot A^T (f - Au(t_k)) \right) \exp(-\alpha (t - t_k)) + \frac{1}{\alpha} e_j \cdot A^T (f - Au(t_k)) .$$

Again, the time steps at which the solution changes can be calculated explicitly and the regularized aISS algorithm can be given as

Algorithm 4. Adaptive Regularized Inverse Scale Space Method

1. **Parameters:** A, f , threshold ≥ 0
 2. **Initialization:** $t_1 = -\log (1 - \alpha / \|A^T f\|_\infty) / \alpha$,
 $p(t_1) = (A^T f) (1 - \exp(\alpha t_1)) / \alpha$, $I_1 = \{i \mid |p_i(t_1)| = 1\}$
- while** $\|Au(t_k) - f\| > \text{threshold}$ **do**
- Compute $u(t_k) = \arg \min_u \{ \|AP_{I_k} u - f\|^2 + \alpha (P_{I_k} u \cdot p(t_k)) \}$ subject to $u(t_k)p(t_k) \geq 0$
 - Obtain t_{k+1} as the minimal time for which (3.8) holds
 - Update the dual variable $p(t)$ via (3.15) with $t = t_{k+1}$
 - Compute $I_{k+1} = \{i \mid |p_i(t_{k+1})| = 1\}$
- end while**
- return** $u(t_k)$
-

4. FURTHER CONVERGENCE ANALYSIS

In the following we provide some additional analysis confirming the favourable properties of the method in typical setups for compressed sensing. We shall assume that

$$(4.1) \quad f = A\hat{u}, \quad P_I\hat{u} = \hat{u}, \quad \hat{u} \cdot e_i \neq 0, i \in I.$$

where \hat{u} is the sparsest solution of $Au = f$. Here I denotes an index set, and P_I the projection onto the elements supported on this index set. Hence, the above condition ensures that I is the index set of nonzero entries of the sparsest solution. In addition, we shall assume a normalization condition on the columns of the matrix A , i.e.,

$$(4.2) \quad \|Ae_i\| = 1, \quad \forall i.$$

First, we can see that each iteration improves the approximation accuracy of the data:

Proposition 2. *The approximation error $\|Au(t) - f\|$ of the inverse scale space flow is strictly decreasing at the times t_k , i.e.,*

$$(4.3) \quad \|Au(t_{k+1}) - f\| < \|Au(t_k) - f\|.$$

Proof. We will prove the above proposition in two steps.

- (1) Show that $\|Au(t_{k+1}) - f\| < \|Au(t_k) - f\|$ if $p(t_k) \notin \partial|u(t_{k+1})|_1$.
- (2) Show that $p(t_k) \notin \partial|u(t_{k+1})|_1$ is always satisfied.

First part: Let us assume that $p(t_k) \notin \partial|u(t_{k+1})|_1$. In this case

$$(4.4) \quad D^{p(t_k)}(u(t_{k+1}), u(t_k)) > 0.$$

Notice that $u(t_{k+1})$ is a minimizer of

$$(4.5) \quad Q(u) = \frac{1}{2}(t_{k+1} - t_k)\|Au - f\|^2 + D^{p(t_k)}(u, u(t_k)),$$

which can easily be verified by confirming that the formula for $p(t_{k+1})$ coincides with the optimality condition of the above functional. Using (4.4) this yields the conclusion

$$\begin{aligned} \frac{1}{2}(t_{k+1} - t_k)\|Au(t_{k+1}) - f\|^2 &< Q(u(t_{k+1})) \\ &\leq Q(u(t_k)) \\ &= \frac{1}{2}(t_{k+1} - t_k)\|Au(t_k) - f\|^2, \end{aligned}$$

and since $(t_{k+1} - t_k) > 0$ we have shown $\|Au(t_{k+1}) - f\| < \|Au(t_k) - f\|$.

Second part: By construction, more specific by the choice of t^{k+1} , there exists an index i such that $|p_i(t_{k+1})| = 1$ and $|p_i(t_k)| < 1$. Let us assume $p_i(t_{k+1}) = 1$, $p_i(t_k) < 1$, and i is, without restriction of generality, the only index at which the value of the subgradient becomes 1 (the negative case is similar and so is the case of multiple indices). We will show that $u_i(t_{k+1}) > 0$ which then (by the characterization of the subdifferential (3.2)) allows the conclusion $p_i(t_k) \notin \partial|u(t_{k+1})|_1$.

Given $p_i(t_{k+1}) = 1, p_i(t_k) < 1$ we know that

$$[A^T(f - Au(t^k))]_i = \frac{1}{t_{k+1} - t_k} (p_i(t_{k+1}) - p_i(t_k)) > 0.$$

Now we can prove $u_i(t^{k+1}) > 0$ by contradiction. If we had $u_i(t^{k+1}) = 0$, we already knew that $u(t_{k+1}) = u(t_k)$, because $u(t_{k+1})$ is determined as the minimizer with nonzero components only on an index set I , which differs from the previous index set of nonzero components only by the index i . Based on our previous calculation, this would mean that also $[A^T(f - Au(t^{k+1}))]_i > 0$. However, $u(t_{k+1})$ is determined as the minimizer of

$$(4.6) \quad \frac{1}{2} \|AP_{I_k}u - f\|^2 + \lambda \cdot u,$$

with Lagrange multipliers λ that enforce the constraint $u_j \geq 0$, if $p_j = 1, u_j \leq 0$, if $p_j = -1$, which means that $\lambda_j \leq 0$ if $p_j = 1, \lambda_j \geq 0$ if $p_j = -1$. Now the optimality condition to (4.6) in the i^{th} coefficient tells us that

$$0 \geq \lambda_i = [A^T(f - Au(t^{k+1}))]_i,$$

which is a contradiction to $[A^T(f - Au(t^{k+1}))]_i > 0$. Therefore, our assumption must have been wrong and thus $u_i(t^{k+1}) > 0$, which means $p_i(t_k) \notin \partial|u(t_{k+1})|_1$. \square

The previous proposition allows us to conclude the finite time convergence of aISS to an ℓ^1 -minimizing solution:

Theorem 2. *Let (u, p) be a solution of the adaptive inverse scale space method as above, then there exists a $K > 0$ such that $t_{K+1} = \infty$ and $u(t)$ is an ℓ^1 -minimizing solution for $t \geq t_K$.*

Proof. Let us denote

$$\begin{aligned} I_1^k &= \{i : p_i(t_k) = 1\}, \\ I_2^k &= \{i : p_i(t_k) = -1\}, \\ I_3^k &= \{i : |p_i(t_k)| < 1\}. \end{aligned}$$

Notice that the solution $u(t_k)$ and therefore the ℓ^2 error $\|Au(t_k) - f\|$ only depends on the index sets I_1^k, I_2^k, I_3^k . If there exists an $l \neq k$ such that $I_1^k = I_1^l, I_2^k = I_2^l, I_3^k = I_3^l$, then obviously $\|Au(t_k) - f\| = \|Au(t_l) - f\|$. However, Proposition 2 shows that this cannot happen, i.e., $\|Au(t_k) - f\| < \|Au(t_l) - f\|$ for $k > l$. Since in finite dimensions there are only finitely many possibilities for I_1^k, I_2^k, I_3^k to be different we can conclude that the method has to converge in a finite number of iterations, i.e., there exists a $K > 0$ such that $t_{K+1} = \infty$.

As we have seen in Theorem 1, $t_{K+1} = \infty$ implies $A^T Au(t_K) = A^T f$ and since in (4.1) we have assumed that f is in the range of A , we obtain $Au(t^k) = f$. To show that $u(t_k)$ indeed is an ℓ^1 -minimizing solution, let \tilde{u} be another solution to

$Au = f$. Then the Bregman distance between \tilde{u} and $u(t_K)$ is

$$\begin{aligned} 0 &\leq D^{p(t_K)}(\tilde{u}, u(t_K)) \\ &= \|\tilde{u}\|_1 - \|u(t_K)\|_1 - \langle p(t_K), \tilde{u} - u(t_K) \rangle \\ &= \|\tilde{u}\|_1 - \|u(t_K)\|_1 - \left\langle \sum_{i=1}^K (t_i - t_{i-1}) A^T (f - Au(t_{i-1})), \tilde{u} - u(t_K) \right\rangle \\ &= \|\tilde{u}\|_1 - \|u(t_K)\|_1 - \left\langle \sum_{i=1}^K (t_i - t_{i-1}) (f - Au(t_{i-1})), \underbrace{A\tilde{u}}_{=f} - \underbrace{Au(t_K)}_{=f} \right\rangle \\ &= \|\tilde{u}\|_1 - \|u(t_K)\|_1, \end{aligned}$$

which shows that $u(t_K)$ is an ℓ^1 -minimizing solution. □

The above proof of Theorem 2 yields finite time convergence but not much information about the complexity needed to reach the desired solution. In a reasonable setup we expect convergence with low complexity, i.e., few iteration steps with small support of the iterates. We shall obtain further information on the complexity by the following analysis. We start with a simple property for only one nonzero entry:

Proposition 3. *Let $|I| = 1$ and let (u, p) be a solution of the adaptive inverse scale space method as above. Then the aISS method converges in one step. If, additionally, $Ae_i \cdot Ae_j < 1$ for $i \neq j$, i.e., if A does not contain two columns which are the same up to a sign, then $u = \hat{u}$ for $t \geq t_1$.*

Proof. Let k be the index of the nonzero entry of \hat{u} . For $t < t_1$ we have

$$\partial_t p_i = (Ae_i) \cdot f = \hat{u}_k (Ae_i) \cdot (Ae_k).$$

From (4.2) and the Cauchy-Schwarz inequality we see

$$p_k(t) = t \hat{u}_k$$

and for $i \neq k$,

$$|p_i(t)| = t |\hat{u}_k| |(Ae_i) \cdot (Ae_k)| \leq t |\hat{u}_k| = |p_k(t)|.$$

In case we have the additional condition $Ae_i \cdot Ae_k < 1$ for $i \neq k$, the above inequality is strict. Hence for $t = t_1$ we have $|p_i(t_1)| < 1$ for $i \neq k$ and $p_k(t_1)$ equals the sign of \hat{u}_k . Consequently, $u(t_1)$ is determined by minimizing $\|Au - f\|$ over all u such that only the k^{th} component is nonzero and has the same sign of \hat{u}_k , thus \hat{u} is the obvious minimizer. Since $A^T(Au - f) = 0$ we obtain $\partial_t p = 0$ for $t \geq t_1$, thus u remains unchanged.

In case we do not have the additional condition $Ae_i \cdot Ae_k < 1$ for $i \neq k$, there may be indices $i \neq k$ at which $u(t_1)$ is allowed to be nonzero. The minimizer of $\|Au - f\|$ under the resulting constraints might not be unique, but for any minimizer we again obtain $A^T(Au - f) = 0$ and thus u remains unchanged for $t \geq t_1$. Notice that in this case the nonuniqueness of $\|Au - f\|$ under the resulting constraints comes from the fact that the ℓ^1 -minimizer is not unique. □

For more than one nonzero entry we need further properties of the matrix A , several of which are regularly used in compressed sensing. The most prominent example is the *restricted isometry property* (RIP) due to Candes and Tao [10], for which equivalence of ℓ^0 and ℓ^1 minimization can be shown. Here we shall use a

weaker property due to Tropp [29], the so-called *exact recovery condition* (ERC), which can be used to show that also in the noisy case the exact support can be reconstructed:

$$(4.7) \quad \|(AP_I)^\dagger Ae_j\|_1 < 1 \quad \forall j \notin I.$$

Using the exact recovery condition we can verify that the inverse scale space method only operates on the support of the exact solution:

Proposition 4. *Let $f = AP_I \hat{u}$ and let condition (4.7) be satisfied. Moreover, let $(u(t), p(t))$ be the solution of the inverse scale space method as above. Then $|p_j(t)| < 1$ for all $j \notin I$ and all $t > 0$.*

Proof. We look for a solution $u = P_I u$ and project the equation

$$\partial_t p = A^T(f - Au) = A^T AP_I(\hat{u} - u)$$

onto I and with the regularity of $(AP_I)^T AP_I$ we find

$$P_I(\hat{u} - u) = (P_I A^T AP_I)^{-1} \partial_t P_I p$$

and hence

$$AP_I(\hat{u} - u) = A(P_I A^T AP_I)^{-1} \partial_t P_I p = ((AP_I)^\dagger)^T \partial_t P_I p.$$

Now let $j \notin I$, then

$$\partial_t p_j(t) = e_j \cdot A^T AP_I(\hat{u} - u) = (Ae_j) \cdot ((AP_I)^\dagger)^T \partial_t P_I p.$$

Since all initial values are zero we can integrate this identity to obtain

$$p_j(t) = ((AP_I)^\dagger Ae_j) \cdot P_I p.$$

Now (4.7) and $\|P_I p\|_\infty \leq 1$ imply $|p_j(t)| < 1$ for all t . □

With (4.7) we can also obtain a result further confirming the optimal behaviour for a very small index set:

Proposition 5. *Assume $|I| = 2$, (4.7), and let (u, p) be a solution of the adaptive inverse scale space method as above. Then $u = \hat{u}$ for $t \geq t_2$.*

Proof. Without restriction of generality let $|\hat{u}_1| \geq |\hat{u}_2| > 0$ be the nonzero elements. We already know that $u_j(t) = 0$ for all $t > 0$ and $j > 2$, thus it suffices to consider the two-dimensional subspace. In the following, let A_i denote the i^{th} column of A . If $\hat{u}_1 > 0$, then for $0 < t < t_1$ we have

$$\begin{aligned} \frac{1}{t} p_1(t) &= [A^T f]_1 = A_1 \cdot f = A_1 \cdot A\hat{u} = A_1 \cdot (A_1 \hat{u}_1 + A_2 \hat{u}_2) \\ &= \underbrace{(A_1 \cdot A_1)}_{=1} \hat{u}_1 + \underbrace{(A_1 \cdot A_2)}_{>-1} \hat{u}_2 \\ &> \hat{u}_1 - \hat{u}_2 \geq 0 \end{aligned}$$

while for $\hat{u}_1 < 0$ we can show in a similar fashion that

$$p_1(t) < 0.$$

Hence, p_1 has the same sign as \hat{u}_1 and by considering the different cases of signs for \hat{u}_1 and \hat{u}_2 one can easily verify that $|p_1(t)| \geq |p_2(t)|$. If equality holds, then it is easy to see that $|\hat{u}_1| = |\hat{u}_2|$ and thus, $u(t_1)$ is obtained by minimizing the residual on the two-dimensional subspace with indices I . Since the signs of $u_i(t_1)$, $i = 1, 2$,

are the same as the signs of \hat{u}_i , we obviously have $u(t_1) = \hat{u}$ and thus $u(t) = \hat{u}$ for all $t \geq t_1$.

If $|p_1(t)| > |p_2(t)|$ the residual is minimized over the one-dimensional subspace with index 1 and sign constraint. It is easy to verify that $u_1(t_1) = 0$ cannot be the minimizer, thus

$$A_1 \cdot (Au(t_1) - f) = 0$$

and consequently p_1 remains constant in (t_1, t_2) . Due to Proposition 4 we must have $|p_2(t_2)| = 1$ and it is easy to check that the sign of p_2 equals the sign of \hat{u}_2 . Hence, the minimization of the residual at time t_2 is carried out over the two-dimensional subspace I with same signs as \hat{u} , which implies $u(t_2) = \hat{u}$ and thus, $u(t) = \hat{u}$ for $t \geq t_2$. \square

5. NUMERICAL RESULTS

In the following we provide some numerical investigations of the adaptive inverse scale space method in different setups and discuss the comparison with OMP.

5.1. Random matrix. To get a better understanding of the behavior of the inverse scale space flow, let us look at the simple case of generating a matrix $A \in \mathbb{R}^{m \times n}$, $n \ll m$, with random values between 0.5 and -0.5 . We normalize each column with respect to the two-norm and generate a signal u_{true} which has random values between -5 and 5 at s random indices, where the sparsity level s is small in comparison to the size n of the signal. The data is generated as $f = Au_{\text{true}}$. Figure 1 shows the aISS iterations as well as u_{true} for an example of $n = 50$, $m = 1000$, and $s = 5$. We can see that on the first seven iterations aISS reconstructs one peak at a time. The first three peaks are indeed peaks of the true solution. The peaks reconstructed in iteration four and five are not part of the true solution. However, in the course of the iteration aISS does find the remaining two true peaks and immediately eliminates the three false peaks in the last iteration as the subgradient of the last missing peak (at index 310) reaches 1 and therefore is included in the support.

For more complicated examples our method changes its support more frequently. It can remove indices from the support not only on the last iteration and might also change more than one index at a time. We expect the ability to arbitrarily change the support to be an advantage over OMP. Before going into a detailed comparison with OMP let us investigate the convergence properties of our method with the related method of linearized Bregman using kicking.

5.2. Comparison with linearized Bregman with kicking. As mentioned in Section 3.2.2 the final result of the linearized Bregman algorithm coincides with the results of our aISS algorithm, if the elastic net parameter δ is chosen large enough. Hence, instead of comparing the two algorithms' results it makes more sense to investigate their convergence properties, particularly the convergence speed. Unfortunately, such a comparison is difficult since it depends on many factors like:

- The sparsity of the ℓ^1 -minimizing solution: The aISS method operates only on a small subset of indices corresponding to nonzero components of the current iterate, while the linearized Bregman method has an explicit update for the whole iterate independent of its sparsity. Therefore, we can expect aISS to perform better the sparser the true solution is.

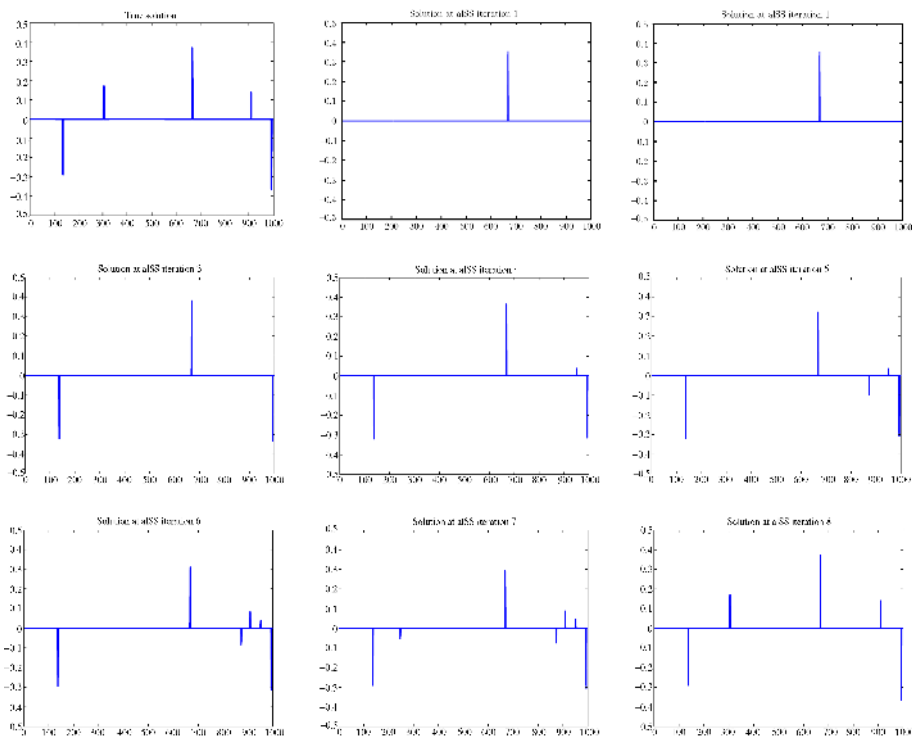


FIGURE 1. Example iteration of the aISS method for a random matrix. The true input signal is shown in the upper left.

- The choice of parameters: The parameters δ and λ need to be chosen for the linearized Bregman method and different parameters yield different convergence speeds. Furthermore, one has to ensure δ to be large enough to guarantee convergence to the ℓ^1 -minimizing solution. Notice that the aISS method does not require any choice of parameter.
- The desired accuracy: While the aISS method converges in a finite number of iterations to the correct subgradient which easily allows to calculate the solution up to machine accuracy, the threshold we choose to determine the convergence of the linearized Bregman method might be critical. Generally, we expect aISS to be faster the higher the accuracy is we require the solution to have.

In the left plot of Figure 2 we illustrate some differences between linearized Bregman with and without kicking as well as the aISS method. We generated a random 200×500 matrix A with entries drawn from a normal distribution and normalized the columns. We generated the data f as $f = A\tilde{u}$ with \tilde{u} being entirely zero except for 25 values being $+1$ or -1 . The comparison shows the decay of the residual $\|Au^k - f\|^2$ over the number of iterations for each method. We can clearly see the advantage of linearized Bregman with kicking over the version without kicking. While the two methods generally perform the same iteration the kicking skipped some parts of the longer stretches in which the linearized Bregman solution does not seem to change. This behavior is stronger for small λ , which is why

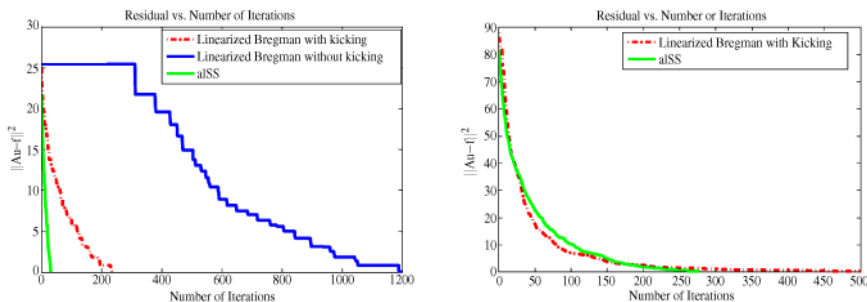


FIGURE 2. Comparison between aISS and linearized Bregman iteration

we intentionally chose $\lambda = 1/600$ to illustrate the staircase-like energy decay of the linearized Bregman method without kicking. The left part of Figure 2 also shows that aISS clearly outperformed both linearized Bregman methods in terms of the decay of the residual per iteration. However, each iteration of the linearized Bregman algorithm is less expensive such that the advantage in speed of aISS was at about 20% for this particular example and this choice of λ .

As mentioned above, the dependencies of the different methods on the sparsity, desired accuracy and parameters makes it difficult to compare the methods. In the right plot of Figure 2 we chose an example of a normalized random 300×500 matrix A with $f = A\tilde{u}$ and \tilde{u} consisting of 100 random $+1$ or -1 entries. Since the signal is less sparse, aISS needs longer to converge. For this example, aISS has a smaller residual for the first iterations, a larger residual for iterations 30 to 175 and finally beats the linearized Bregman algorithm by jumping down from a residual of 0.27 at iteration 276 to a residual of 10^{-28} at iteration 277. The linearized Bregman algorithm needs 6294 iterations to get down to a residual of 10^{-26} , however, both algorithms need about 2 seconds to obtain this accuracy. If, for instance, an accuracy of 10^{-5} is sufficient, linearized Bregman only needs 0.32 seconds to converge in this particular example.

In general, the best choice of an ℓ^1 -minimization algorithm will depend on the particular application, matrix and size of the problem as well as the type, desired accuracy and sparsity of solution. Our aISS method is particularly well suited for problems with very sparse solutions for which a high accuracy is desirable.

5.3. Comparison with OMP. In this subsection we compare aISS and OMP in two steps. First, we will describe the problem of monotonic increase of the support of OMP and present an algorithm to construct a sensing matrix A and a signal f for which OMP fails. Second, we will compare aISS and OMP on different types of matrices including random matrices, combined wavelet matrices, and ill-conditioned matrices arising in dynamic positron emission tomography (PET) to see how frequently the problem setup we present in the next subsection occurs in realistic, practical settings.

5.3.1. A counterexample for OMP. As discussed before, a major difference between OMP and aISS is that OMP never decreases the support of the solution and (in the terminology of aISS) resets the dual variable/subgradient after each iteration, whereas aISS is able to decrease its support and continuously evolve the subgradient, taking the information of all previous iterations into account. In this subsection we

will show that this difference can have a major effect on the reconstruction results leading to arbitrarily nonsparse results of OMP while aISS can still recover the exact solution.

We can construct an example for which OMP fails as follows: Assume we have data f which can be written as $f = c(v_1 + v_2)$ for two normalized vectors v_1, v_2 and a constant c . In the end, we will include v_1 and v_2 in our matrix A , since this will guarantee the sparsest solution of $Au = f$ to be 2-sparse. We start the construction of A by choosing a small ϵ , setting $A^0 = \emptyset$, and define an $r_0 = f$, which will correspond to the residual in the construction. Let $f \in \mathbb{R}^n$, then we iteratively choose for $i = 1, \dots, n$,

$$\begin{aligned} (1) \quad & A^i = [A^{i-1}, r_{i-1} - \epsilon e_i], \\ (2) \quad & r_i = f - A^i((A^i)^\dagger f), \end{aligned}$$

where $(A^i)^\dagger$ denotes the generalized inverse of the current matrix A^i . The first step takes the current residual and deflects it by a small ϵ in the direction of the i^{th} unit normal vector. This will make OMP select this vector in the i^{th} step since for ϵ small enough, the correlation of this vector to the current residual will be maximal. However, the small disturbance will lead to OMP not having converged yet. Therefore, we compute the next residual r_i that will come up in the OMP algorithm and again provide a column in A^i that has a very high correlation to this residual but does not enable the method to solve $A^i u = f$ exactly. Iteratively, we construct n such vectors. Note that n is the least sparse vector one can get for the description of $f \in \mathbb{R}^n$, since any additional vector would automatically be linearly dependent. Finally, we add v_1 and v_2 as the last two columns to obtain the final $A = [A^n, v_1, v_2]$, which we normalize afterwards. Although v_1 and v_2 are sufficient to describe f they will likely not have the highest correlation to any residual of OMP.

Figure 3 shows some iterations of the OMP algorithm on an A constructed as described above with $n = 50$ and $v_1 = e_n, v_2 = e_{n-1}, f = 0.5v_1 + 0.5v_2, \epsilon = 0.15$. As we can see OMP indeed adds one component after the other until at the 50th iteration it finally has the maximum number of linearly independent vectors in \mathbb{R}^n and can therefore reconstruct f exactly. The OMP answer is as nonsparse as possible although the input signal was 2-sparse (which is the sparsest possible without being reconstructed in 1 iteration by either of the two methods aISS or OMP).

Now let us take a look at the iteration of aISS for the same example. Figure 4 shows not only the solution at each iteration but also the corresponding subgradient since this is crucial for understanding the difference between the two methods.

We can see that the first three iterations coincide with the OMP method, and components corresponding to a high correlation with the residual are added to the set of nonzero elements. However, the subgradient shows that aISS “sees” that the last two components in the matrix A also have a good correlation to the signal f . Their correlation is not as high as for the first columns of A which is why neither aISS nor OMP added them immediately. However, OMP basically resets the subgradient at each iteration and includes the vector with the highest correlation to the residual. aISS on the contrary, does not reset the subgradient and keeps adding to the correlation of the last 2 vectors until, at iteration 4, the subgradient of components 51 and 52 hits 1, which immediately leads to the right,

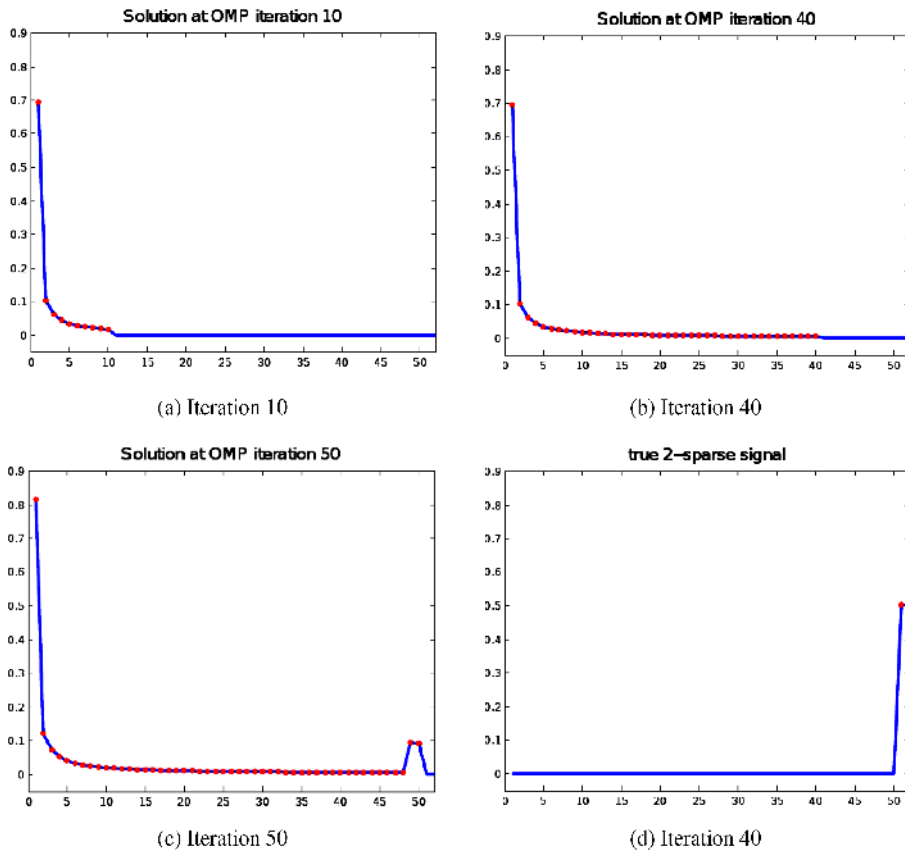


FIGURE 3. Iterates of OMP on constructed example. Dots indicate the coefficients at which the current solution is nonzero.

2-sparse answer. aISS converged in only 4 iterations to the correct answer while OMP converged in 50 iterations to the least sparse answer possible.

5.3.2. *Random matrices and combined wavelet basis.* In this subsection we will investigate how aISS and OMP compare on noise free data. The criteria for our comparison will be:

- (1) frequency of exact recovery of the sparsest signal;
- (2) sparsity of the solution each algorithm found, i.e., number of nonzero elements $|u|_0$;
- (3) number of iterations each algorithm took;
- (4) runtime of each algorithm.

The tests will be based on cases, where neither of the two algorithms is guaranteed to converge to the sparsest solution (for theoretical guarantees for this kind of convergence see, for instance, [23]). Notice, that aISS will always converge to the ℓ^1 -minimizing solution of $Au = f$. The criteria of frequency of exact recovery as well as sparsity of the solution are therefore rather based on the question of whether the ℓ^1 -minimizing solution coincides with the ℓ^0 -minimizing solution than a convergence/quality property of our algorithm in particular. The examples where

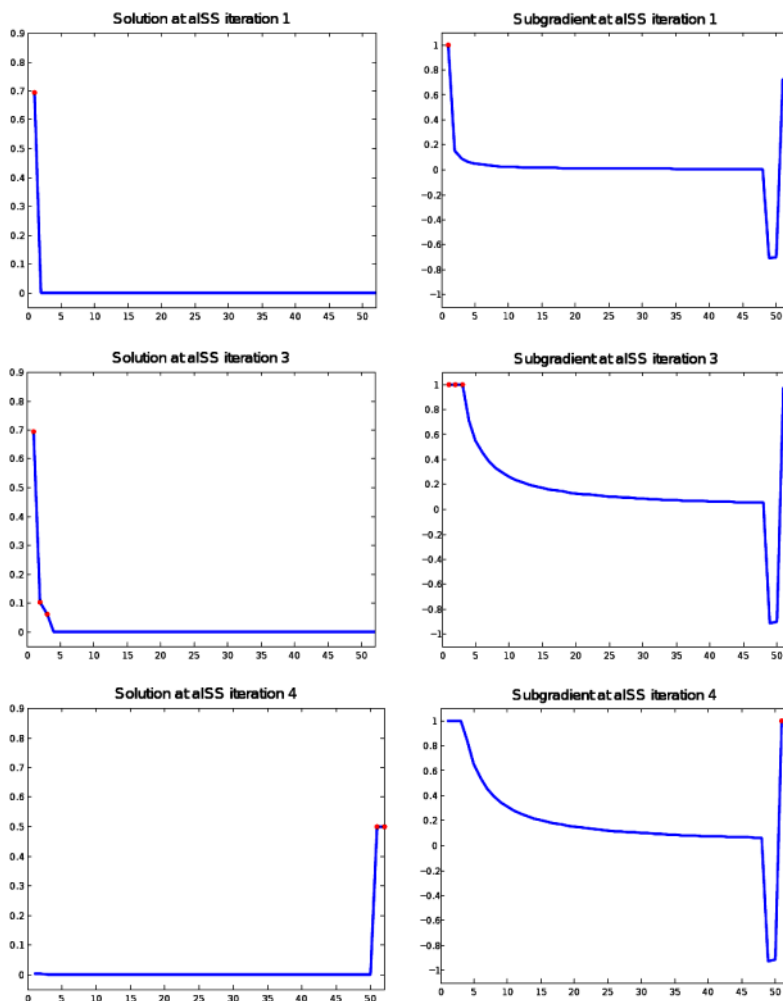


FIGURE 4. Iterates of aISS on constructed example. Dots indicate the coefficients at which the current solution is nonzero.

aISS does not reconstruct the sparsest solution are therefore a violation of the requirements we need for the ℓ^1 -minimizing solution to be the sparsest solution.

In the first experiment we generate a matrix $A \in \mathbb{R}^{128 \times 512}$ with random entries between -0.5 and 0.5 . Then, we normalize each column of A and generate a signal u_{true} also with random values between -0.5 and 0.5 at s random indices. s is the sparsity level of the true, sparsest solution. We vary s from 10 to 60, run each algorithm 50 times per s , and record the comparison metrics described above. Figure 5 shows the results among all sparsity levels.

We can see that the greedy approach to ℓ^0 -minimization works much better in this example. The frequency of exact recovery is higher, and even for the cases where the sparsest solution is not recovered exactly, the sparsity of the OMP solution is much better than for the aISS solution. Furthermore, OMP obtained its results in fewer iterations and less runtime.

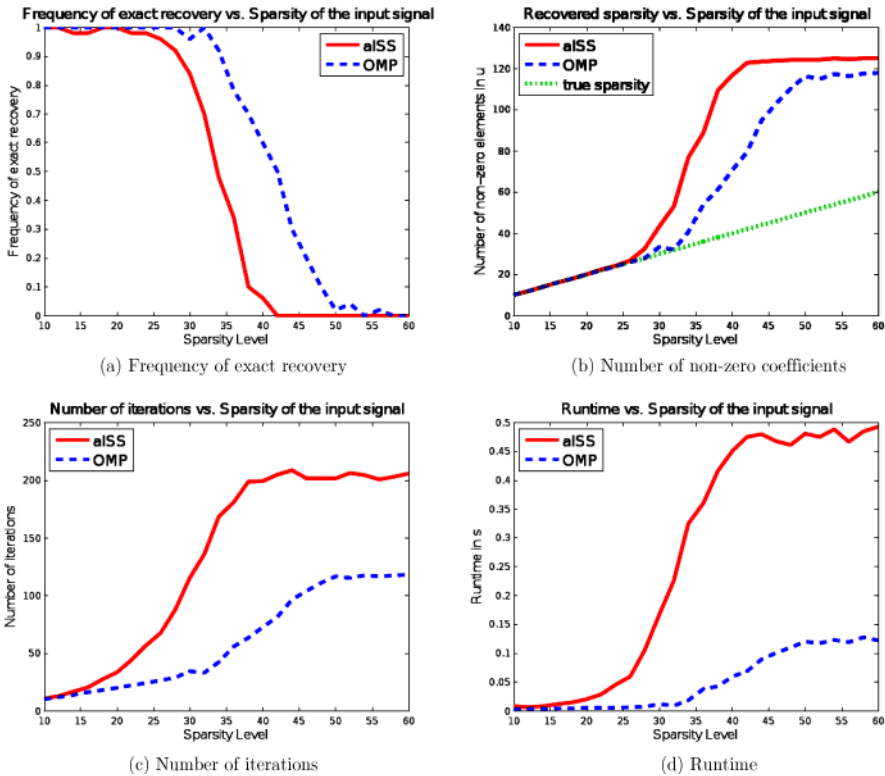


FIGURE 5. Comparison of OMP and aISS on random matrices and input signals with random values. The plots show the comparison matrices with respect to the sparsity level of the input signal.

Despite the very good results OMP gave in this example, we will see that this is not the case in general. In our second example, we create the random matrix A in the same fashion as in the previous experiment. The only change is that we generate the signal u_{true} as a random sign function at s coefficients, i.e., we set s values in u_{true} randomly to -1 or $+1$. As we can see in Figure 6 this changes the results dramatically.

In this case, aISS clearly outperforms OMP regarding the quality of the results, yielding better frequency of recovery and sparser solutions. It is interesting to see that the aISS recovery frequency stays almost the same as in the previous example, whereas the recovery frequency of OMP dropped significantly. Note that the number of iterations of OMP and aISS is very similar up to a sparsity of 32 of the input signal. The runtime is almost the same up to a sparsity of 28. After that, aISS pays for the higher accuracy and higher sparsity with more computational afford; however, even for the most complex example the runtime is below 0.5 seconds using a laptop with 2Ghz dual core processor and 3GB memory.

We obtain similar results on more structured, over-complete basis like the combination of Matlabs “Haar” and “Daubechies4” basis for a level 6 decomposition. With A being generated as the normalized combination of those two bases, again using 128 to be the number of rows of A , we choose a signal u_{true} also with random

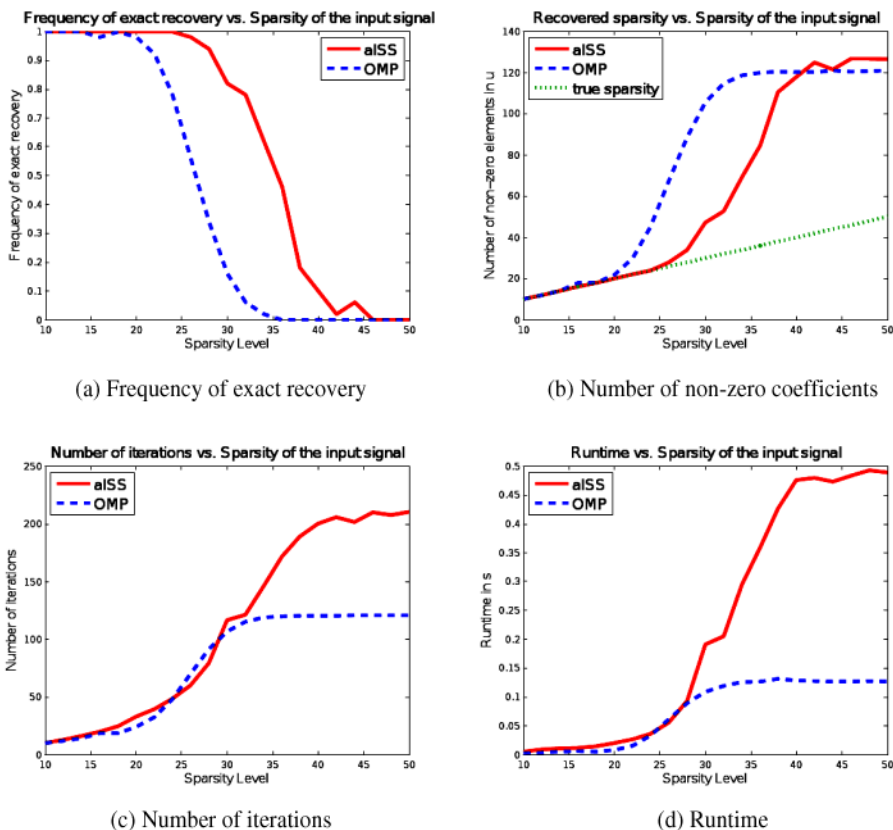


FIGURE 6. Comparison of OMP and aISS on random matrices and input signals with random $+1$ and -1 values. The plots show the comparison metrics with respect to the sparsity level of the input signal.

values between -0.5 and 0.5 at s random indices. Figure 7 shows the comparison of OMP and aISS in this test case.

First, we can see that this example was more challenging for both algorithms yielding a lower frequency of recovery at earlier iterations in comparison to the random matrices. Again, aISS outperforms OMP in terms of frequency of exact recovery and sparsity of the recovered signal. In this case, even the number of iterations is similar, whereas the runtime is similar up to iteration 20, then increasing stronger but still being relatively fast.

As mentioned earlier, aISS finds the ℓ^1 -minimizer which might not be the sparsest solution. Thus, we could also compare two other cases: If among all examples mentioned above, we compare OMP and aISS only on the cases where both methods reconstruct the sparsest solution exactly, OMP gives the correct result faster only needing about 0.0045 seconds whereas aISS needs 0.027 seconds on average. This difference in speed is also due to the average number of iterations needed to find the solution which is 16.2 for OMP and 30.3 for aISS. Comparing the methods only for the cases where the ℓ^1 -minimizing solution coincides with the ℓ^0 -minimizing

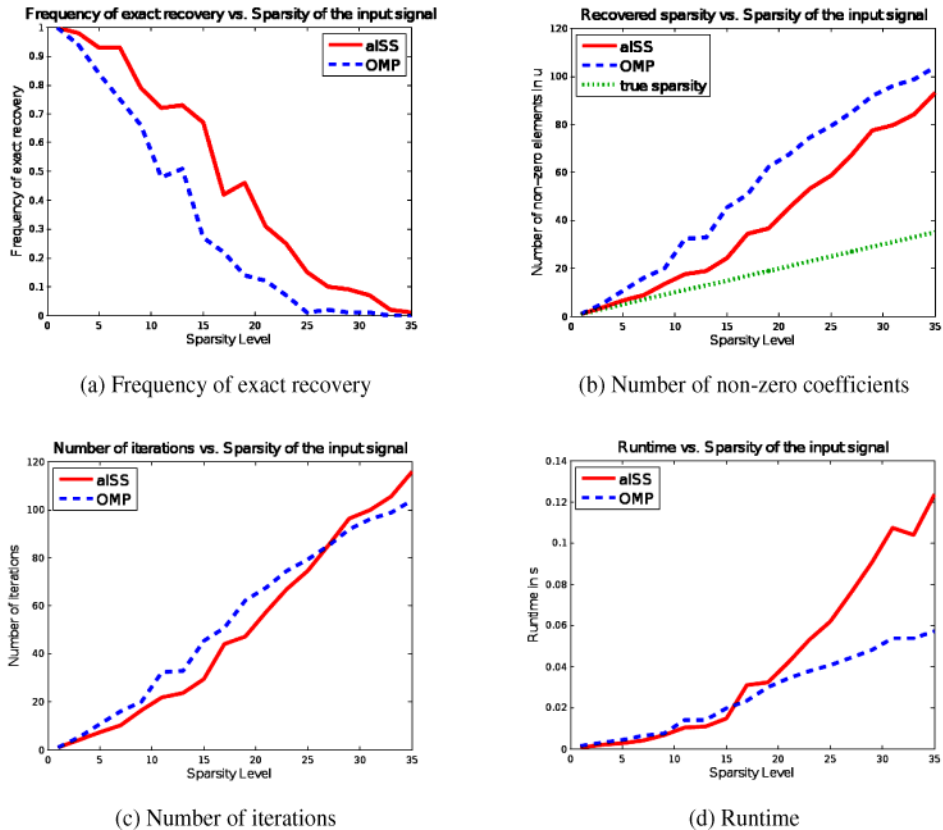


FIGURE 7. Comparison of OMP and aISS on a matrix of combined “Haar” and “Daubechies4” wavelet basis and input signals with random values. The plots show the comparison metrics with respect to the sparsity level of the input signal.

solution we have a sparsest signal recovery rate of 100% for aISS and about 75.8% for OMP. In these examples the average OMP solution is much more dense with an average number of nonzero coefficients of 32.4 opposed to 17.3 for the aISS solutions.

To also compare OMP and aISS on matrices A used in practical applications we will discuss temporal basis functions for dynamic PET in the next subsection.

5.3.3. Temporal basis functions for dynamic PET. In [28] an exponential basis operator has been introduced in order to improve dynamic Positron Emission Tomography (PET) images. For applications such as myocardial perfusion quantification (see for instance [1]) it is a standard assumption that the measured dynamic signal is a composition of a so-called input function (which we assume to be known) and a Laplace-convolution of that input function with a specific exponential function. In this short computational example we therefore intend to solve the inverse problem

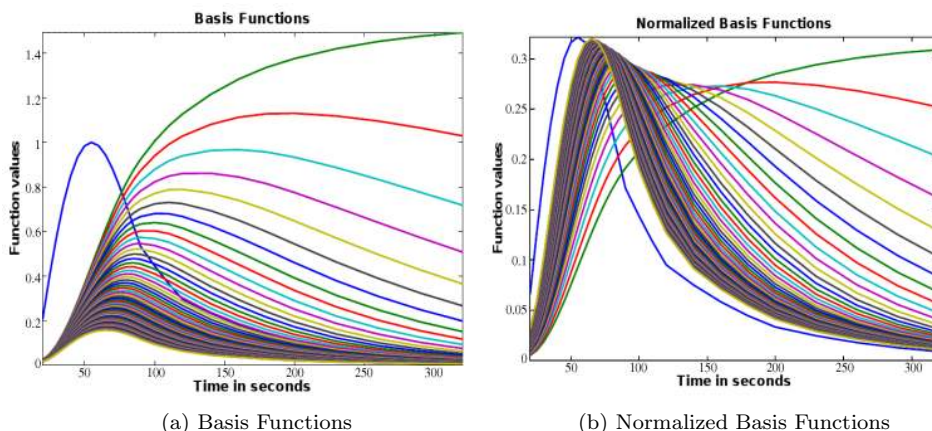


FIGURE 8. The columns of A as defined in this section. The left-hand side shows the columns, representing the basis functions (5.2). The right-hand side shows the normalized columns with respect to the 2-norm.

$Au = f$ with the operator A given as the linear combination

$$(5.1) \quad (Au)(t) := u_0 h(t) + \sum_{n=1}^N u_n \tilde{b}_n(t),$$

of basis functions $\tilde{b}_n(t)$ defined as

$$(5.2) \quad \tilde{b}_n(t) := \int_0^t h(\tau) \exp(-b_n(t - \tau)) d\tau,$$

for a given positive input function $h : [0, T] \rightarrow \mathbb{R} \cup \{\infty\}$ and a given vector $b = (b_n)_{n \in \{1, \dots, N\}}$ containing nonnegative real values. For the sake of simplicity, we focus on a one-dimensional setting; to simulate input data realistically, we sample the time $t \in [0, T]$, with $T = 320$ s, at 26 discrete points $\{20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 100, 110, 120, 140, 160, 180, 200, 230, 260, 290, 320\}$. The basis functions \tilde{b}_n were computed on a finer grid via a simple Euler scheme, for $N = 61$ given values $b_n \in [0, 6]$ with step size 0.1, and were subsequently sampled at the discrete temporal points. The underlying function h is defined as $h(t) := \frac{t}{64} \exp\left(-\frac{t^2}{128}\right)$ and normalized with respect to the $\|\cdot\|_\infty$ -norm. Hence, we obtain a fully discrete matrix $A \in \mathbb{R}^{26 \times 62}$, for which its columns (which are the discrete analogue of the basis functions \tilde{b}_n) can be seen in Figure 8. As in the previous examples we are going to normalize the columns of A in order to weight the basis functions correctly and not to distort reconstructions. From Figure 8 it is already clear that the considered matrix is very ill-conditioned, since the columns appear to be very similar to each other.

TABLE 1. Comparison of aISS and OMP for the matrix A as defined in (5.1) with normalized columns in terms of runtime, sparsity of u (ℓ^0), ℓ^1 -norm, error to u_{true} , standard-deviation between Au and g and standard-deviation between Au and f . All computations have been made on a laptop with a 2.53 GHz dual core processor and 4 GB memory.

Noise Level	$\sigma = 0$		$\sigma = 0.0075$		$\sigma = 0.0125$		$\sigma = 0.03$	
Method	aISS	OMP	aISS	OMP	aISS	OMP	aISS	OMP
Runtime	0.004041	0.0008346	0.004103	0.0005302	0.004306	0.0005242	0.004577	0.0005098
ℓ^0 -norm	2	12	2.27	4.971	2.099	4.611	2.073	4.152
ℓ^1 -norm	2.201	3.198	2.199	3.473	2.198	3.378	2.189	3.217
$\frac{\ u - u_{\text{true}}\ _{\ell^1}}{\ u_{\text{true}}\ _{\ell^1}}$	0	1.607	0.3493	1.744	0.5658	1.739	0.9673	1.721
$\ Au - g\ _{\ell^2}$	0	0	0.002058	0.003509	0.00364	0.005829	0.009601	0.01206
$\ Au - f\ _{\ell^2}$	0	0	0.007254	0.006916	0.01215	0.01158	0.02939	0.02789
No. of iter.	22	12	22.58	4.971	23.77	4.611	25.11	4.152

As described above it is very natural to assume the exact signal g to be a composition of the function h and one single basis function. We therefore define an example such that the true coefficients are given as the vector

$$(5.3) \quad u_{\text{true}}^j := \begin{cases} \approx 0.931 & \text{if } j = 0, \\ \approx 1.270 & \text{if } j = 15, \\ 0 & \text{else,} \end{cases}$$

and the exact data $g = Au_{\text{true}}$. Moreover, we compute a set of noisy datum f , for which we disturb g with normal-distributed noise with mean zero and standard-deviation σ . In Table 1 we have listed the results of several computations with metric values similar to the ones of the previous comparisons. We have compared the runtime, the total number of non-zero coefficients, the ℓ^1 -norms as well as a normed ℓ^1 -difference between u and u_{true} (with u denoting the reconstruction), and the standard-deviations between Au and g and Au and f , respectively, for different noise levels σ . The results are average values over 1000 computations for each value of σ . In the case of $\sigma \neq 0$ we have stopped the aISS- and OMP-computations according to the discrepancy principle that if the standard-deviation between Au and f is below σ the computation is stopped.

It can be seen that for any σ the OMP algorithm needs less iterations and therefore less runtime than the aISS algorithm. However, regarding the quality of the results the aISS outperforms OMP for each σ -value, which we would also expect since the matrix is highly ill-conditioned. In the noise-free case aISS perfectly recovers the two desired coefficients after 21 iterations, while OMP recovers 12 coefficients in order to somehow approximate the exact data without reconstructing a sparse solution. In the presence of noise and by applying the discrepancy principle (which might not guarantee the algorithm to stop at the iteration that produces the best result) we see that aISS is not always computing the sparsest approximation, since the average ℓ^0 -value is larger than two. But in comparison to OMP the results are significantly closer to two-sparsity as the results with OMP. The ℓ^1 -

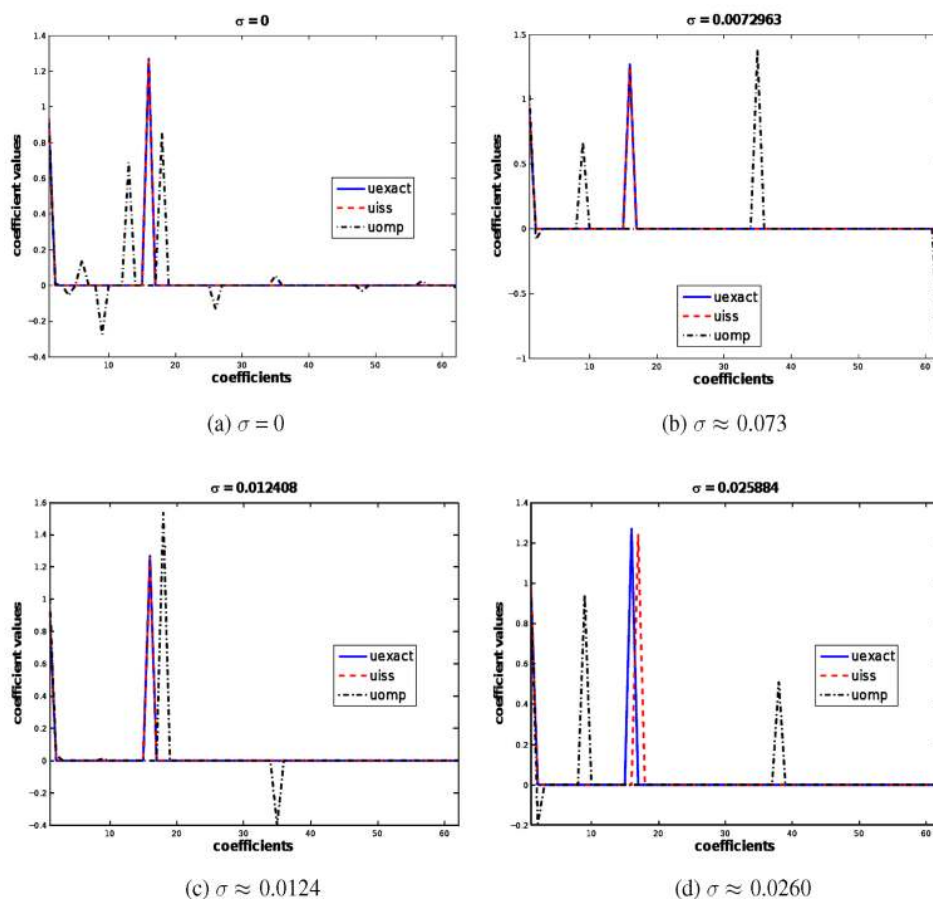


FIGURE 9. Comparison of particular aISS and OMP reconstructions with the true coefficients in the noise-free (Figure 9(a)) and in the noisy case (Figure 9(b)–9(d)), for varying standard deviation σ .

norms of the computed aISS-solution are closer to the ℓ^1 -norm of the true signal, and the weighted ℓ^1 -norm between reconstruction and exact solution is also lower for aISS- than for OMP-computations. In addition, the standard deviation of Au to $Au_{\text{true}} (= g)$ is smaller in the case of aISS than for OMP; the OMP solutions better approximate the noisy data f .

Moreover, we have plotted exemplary reconstructions in Figure 9 for particular noisy functions f that can be seen in Figure 10. For moderate noise aISS is able to recover the true support of the exact solution. Note that there is no additional positivity constraint in the computation of aISS. In the presence of more severe noise neither OMP nor aISS can obtain the true support, which surely is a result of the severely ill-conditioned matrix A , but might also be a consequence of the early stopping due to the discrepancy principle.

Finally, we also run several tests for Algorithm 4, computing a solution of the regularized inverse scale space flow (1.11). We compute the average values over 1000

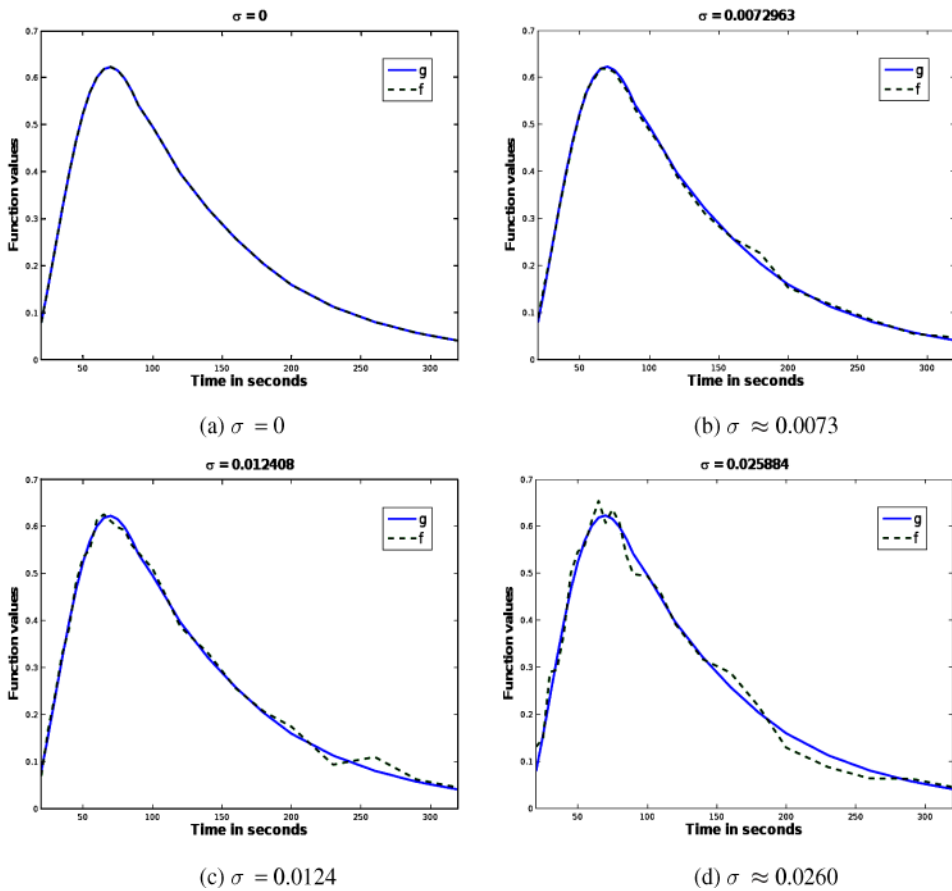


FIGURE 10. The function g and different noisy functions f , for which the noise attains standard deviation σ .

reconstructions for α varying between 0.001 and 1, for different noise levels with standard deviation σ . In Figure 11 we plot almost the same attributes as we have compared in Table 1 (except for the standard deviation between Au and f , which is similar to the standard deviation between Au and g), for varying α and the different noise levels. Obviously the runtime as well as the number of iterations decreases for increasing α , which seems to be very natural since a more regularized solution should become sparser. This is indeed the case if we take a look at Figure 11(b). Moreover, the ℓ^1 -norm (which is ≈ 2.201 for u_{true}) is monotonically decreasing for increasing α , which we also would expect. The weighted ℓ^1 -difference reveals interesting insights, since for increasing α in the noiseless case the difference first increases steeply (up to an α -value of about 0.06) and then increases only slowly; in case of noisy data, the difference decreases first down to an optimal α , which seems to be somewhere in between 0.001 and 0.05, and increases again afterwards. The standard deviation between Au and g is almost monotonically increasing, which is not surprising but rather expectable.

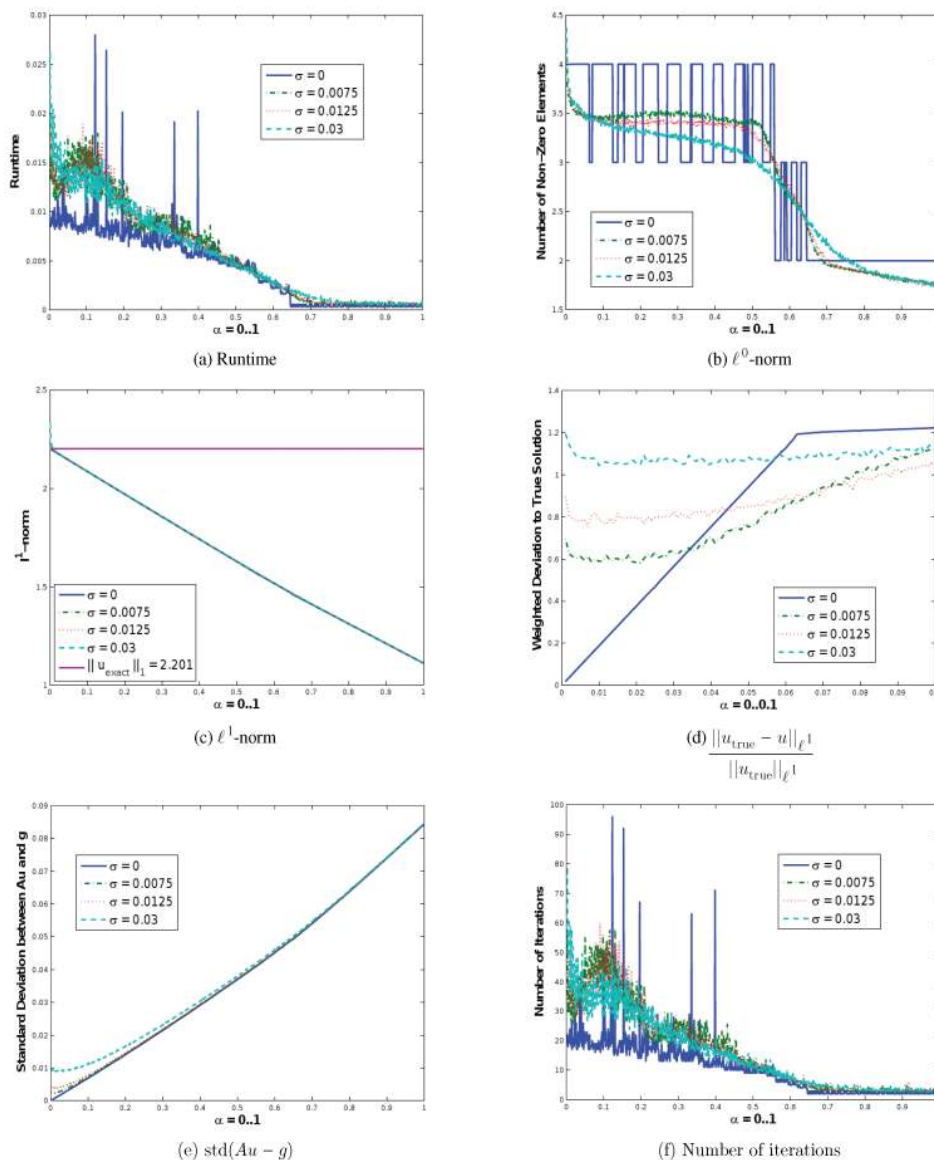


FIGURE 11. The average attributes runtime, ℓ^0 -norm, ℓ^1 -norm, weighted ℓ^1 -deviation between u and u_{true} , standard deviation between Au and g and the number of iterations, over 1000 computations per α - and σ -value, for $\alpha \in [0.001, 1]$ and $\sigma \in \{0, 0.0075, 0.0125, 0.03\}$. All computations have been made on a computer with 2.83 GHz quad core processor with 8 GB memory.

6. CONCLUSIONS & OUTLOOK

We have investigated inverse scale space methods and proposed a new algorithm for solving the inverse scale space flow of ℓ^1 -regularization, which converges in finite time by iteratively solving low-dimensional nonnegative least squares problems, and in addition bears an interesting resemblance to the well-known orthogonal matching pursuit method. Several convergence properties have been shown and extensive numerical comparisons to the OMP algorithm have been presented. The new adaptive ISS method seems to provide robust and qualitatively better results, in particular, for ill-conditioned problems, while still being very efficient.

In future research we will extend the aISS model to regularization functionals other than ℓ^1 and also investigate inverse scale space flows corresponding to data fidelity terms different from ℓ^2 .

REFERENCES

1. M. Benning, T. Kösters, F. Wübbeling, K. Schäfers, and M. Burger, *A nonlinear variational method for improved quantification of myocardial blood flow using dynamic H215O PET*, IEEE Nuclear Science Symposium Conference Record, 2008, pp. 4472–4477.
2. T. Blumensath and M. Davies, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmonic Anal. **27** (2009), 265–274. MR2559726 (2010i:94048)
3. M. Burger, *A note on sparse reconstruction methods*, J. Phys. Conference Series, no. 012002, 2008.
4. M. Burger, K. Frick, S. Osher, and O. Scherzer, *Inverse total variation flow*, SIAM Multiscale Modelling and Simulation **6** (2007), 366–395. MR2338487 (2008h:35212)
5. M. Burger, G. Gilboa, S. Osher, and J. Xu, *Nonlinear inverse scale space methods*, Comm. Math. Sci. **4** (2006), 179–212. MR2204083 (2006i:35177)
6. M. Burger and S. Osher, *A guide to TV zoo*, preprint.
7. M. Burger, E. Resmerita, and L. He, *Error estimation for Bregman iterations and inverse scale space methods in image restoration*, Computing **81** (2007), 109–135. MR2354192 (2008k:94002)
8. J. Cai, S. Osher, and Z. Shen, *Convergence of the linearized Bregman iteration for ℓ_1 -norm minimization*, Math. Comp. **78** (2009), 2127–2136. MR2521281 (2010k:65111)
9. ———, *Linearized Bregman iterations for compressed sensing*, Math. Comp. **78** (2009), 1515–1536. MR2501061 (2010e:65086)
10. E. J. Candes and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory **51** (2004), 4203–4215. MR2243152 (2007b:94313)
11. ———, *Near-optimal signal recovery from random projections: universal encoding strategies*, IEEE Trans. Inform. Theory **52** (2004), 5406–5425. MR2300700 (2008c:94009)
12. A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet schemes for elliptic operator equations—convergence rates*, Math. Comp. **70** (2001), 27–75. MR1803124 (2002h:65201)
13. ———, *Adaptive wavelet methods II—beyond the elliptic case*, Found. Comput. Math. **2** (2002), 203–245. MR1907380 (2003f:65212)
14. S. Dahlke, W. Dahmen, and K. Urban, *Adaptive wavelet methods for saddle point problems—optimal convergence rates*, SIAM J. Numer. Anal. **40** (2002), 1230–1262. MR1951893 (2004g:42039)
15. I. Daubechies, M. DeFrise, and C. DeMol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Commun. Pure Appl. Math. **57** (2004), 1413–1457. MR2077704 (2005k:65106)
16. D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory **52** (2006), 1289–1306. MR2241189 (2007e:94013)
17. D. L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization*, Proc. Natl. Acad. Sci. USA **100** (2003), 2197–2202. MR1963681 (2004c:94068)

18. D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, *Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit (stomp)*, Tech. report, Stanford Univ., 2006, Stat. Dept. Tech. Rep. 2006-02.
19. I. Ekeland and R. Temam, *Convex analysis and variational problems*, corrected reprint edition ed., SIAM, Philadelphia, 1999. MR1727362 (2000j:49001)
20. S. G. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Transactions on Signal Processing **12** (1993), 3397–3415.
21. Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations: The fifteenth Dean Jacqueline B. Lewis memorial lectures*, American Mathematical Society, 2001. MR1852741 (2002j:43001)
22. D. Needell and J. A. Tropp, *Cosamp: Iterative signal recovery from incomplete and inaccurate samples*, Applied and Computational Harmonic Analysis **26** (2009), 301–321. MR2502366 (2010c:94018)
23. D. Needell, J. A. Tropp, and R. Vershynin, *Greedy signal recovery review*, Proc. 42nd Asilomar Conf. Signals, Systems and Computers, 2008.
24. D. Needell and R. Vershynin, *Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit*, IEEE J. Sel. Topics Signal Process **4** (2009), 310–316.
25. S. Osher, M. Burger, D. Goldfarb, and J. Xu and W. Yin, *An iterative regularization method for total variation-based image restoration*, SIAM Multiscale Model. Simul. **4** (2005), 460–489. MR2162864 (2006c:49051)
26. S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast linearized Bregman iteration for compressive sensing and sparse denoising*, Commun. Math. Sci. **8** (2010), 93–111. MR2655902 (2011c:90098)
27. Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers, 1993, pp. 40–44.
28. A. J. Reader, J. C. Matthews, F. C. Sureau, C. Comtat, R. Trébossen, and I. Buvat, *Fully 4D image reconstruction by estimation of an input function and spectral coefficients*, IEEE Nuclear Science Symposium Conference Record, 2007, pp. 3260–3267.
29. J. A. Tropp, *Just relax: Convex programming methods for identifying sparse signals in noise*, IEEE Trans. Inform. Theory **52** (2006), 1030–1051. MR2238069 (2007a:94064)
30. J. A. Tropp and A. C. Gilbert, *Signal recovery from random measurements via orthogonal matching pursuit*, IEEE Trans. Inf. Theory **53** (2007), 4655–4666. MR2446929 (2009h:94042)
31. Y. Wang and W. Yin, *Sparse signal reconstruction via iterative support detection*, SIAM J. Imaging Sci. **3** (2010), 462–491. MR2679436 (2011h:94018)
32. W. Yin, *Analysis and generalizations of the linearized Bregman method*, SIAM J. Imaging Sciences **3** (2010), no. 4, pp. 856–877. MR2735964 (2011j:68172)
33. W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing*, SIAM J. Imaging Sci. **1** (2008), 143–168. MR2475828 (2010f:90170)

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER, INSTITUT FÜR NUMERISCHE UND ANGEWANDTE MATHEMATIK, EINSTEINSTR. 62, D 48149 MÜNSTER, GERMANY
E-mail address: martin.burger@wwu.de

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER, INSTITUT FÜR NUMERISCHE UND ANGEWANDTE MATHEMATIK, EINSTEINSTR. 62, D 48149 MÜNSTER, GERMANY
E-mail address: m.moeller@gmx.net

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER, INSTITUT FÜR NUMERISCHE UND ANGEWANDTE MATHEMATIK, EINSTEINSTR. 62, D 48149 MÜNSTER, GERMANY
E-mail address: martin.benning@wwu.de

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA LOS ANGELES. PORTOLA PLAZA, LOS ANGELES, CALIFORNIA 90095
E-mail address: sjo@math.ucla.edu