An Overview on Algorithms for Sparse Recovery

Massimo Fornasier and Steffen Peter

Abstract Sparsity is a very powerful prior for the identification of signals from noisy indirect measurements. The recovery of the signal is usually performed by suitable linearly constrained optimizations with additional sparsity enforcing barriers. Depending on the specific formulation, one can produce a variety of different algorithms. In this Chapter the numerical realizations of such linear and nonlinear programs are reviewed and compared with respect to different noisy scenarios. With the intention of providing a useful guide to users, we provide a sound mathematical introduction with rigorous derivations and proofs of convergence of all algorithms and we illustrate in detail their behavior by extensive numerical experiments.

1 Introduction: The World is Sparse

Often in real life problems, for instance in remote sensing [REFERENCE TO OTHER CHAPTERS], one cannot dispose of a direct observation of the object of interest, but only of indirect measurements. It is often the case also that such measurements are way fewer than necessary to completely describe the object of interest, because of technical limitations, costs of acquisition or even hazard of the sampling procedure. In this case we are facing an ill-posed problem, as the solution is not unique or does not depend continuously from the data. In order to *regularize* the recovery, some additional information, called *priors* in the statistical literature as they correspond to a conditional probability, are needed. One of the most powerful and effective prior, which gained tremendous attention in the signal processing community in past two decades, both from mathematicians and engineers, is given by the assumption that a signal can be represented somehow *sparsely* in terms of a given basis. The success of this assumption is due to evidence that it is eventually not so difficult, with sometimes just a bit of creative effort, to define a proper transformation to turn nearly any reasonable signal into a sparse one. Let us make some easy to grasp examples. It is common experience that music is pervasive and filling. Nevertheless, when we express it in terms of its score, we all know that not

Massimo Fornasier
Faculty of Mathematics, Technische Universität München, Boltzmannstrasse 3, 85748 Garching, Germany e-mail: massimo.fornasier@ma.tum.de

Steffen Peter
Faculty of Mathematics, Technische Universität München, Boltzmannstrasse 3, 85748 Garching, Germany e-mail: steffen.peter@ma.tum.de

all the notes are simultaneously appearing, but they are rather a sequence in time. Hence, at each instant of time only few frequencies (corresponding to a note and its harmonics) are actually active. This means that music is intrinsically sparse. Described in mathematical terms, a music interpreted as a function of time will have a sparse short-time Fourier transform (often represented in the form of the spectrogram). Similarly, it is common experience to sense images as a complicated information, containing many details from which we often intend to extract meanings. However, for many images of natural or man-made objects, one can roughly describe the image in a simplified form, by naming its connected components in terms of relatively uniform color or gray level and by assigning to each of these components one single color or gray level value. Hence, from a rather sophisticated signal, we distilled a succinct description of it. The mathematical transformation that does such a reduction is as simple as it gets: one just takes the gradient of the image intepreted as a mathematical function of its points. The limitation in defining sparse transformations of signals is really only the one of the creativity. Let us dare to associate a sparse structure to something as abstract as the taste of people about films. It is common experience to meet with friends and watch together a movie. Often, knowing our friends, we can guess whether a movie will be of their taste as well. That's precisely what certain movie renting companies do when they provide expert suggestions to customers for their choice of a movie. If we numerically express (say from 1 to 10) the possible appreciations of customers for a certain collection of movies into a matrix, we should expect to find some correlations between rows or columns indicating that certain groups of people (of same education, age, gender etc.) tend to like similar movies. Mathematically speaking we expect such a matrix to have low rank or to have a sparse spectrum (in terms of singular values). Social dynamics is a complicated superposition of interactions which are iterated in time. We may dare to describe mathematically the evolution of a society as a dynamical system. It is also hope that societies can be fair and stabilize themselves on certain socially balanced equilibria. Unfortunately we all know that this is not always the case and the human community has always advocated eventually the action of a government. However, governments usually dipose of limited resources (usually the taxes of the citizens of the past year) and with those few they need to take parsimonious actions to steer the societies towards a better good. One can prove mathematically that, for certain types of dynamical systems, sparse controls can always stabilize the system, showing once again the powerful machinery of sparse representations.

The identification of the elevation of buildings in an urban area from remote sensing, customers preferences from minimal surveys, a medical image from MRI data, or the proper government action to improve the economy are all eventually formulable as solutions of a sparse optimization problem with linear or nonlinear constraints. This Chapter is an introduction to methods recently developed for performing numerical optimizations with linear model constraints and additional sparsity conditions to solutions, i.e., we expect solutions which can be represented as sparse vectors with respect to a prescribed basis. Such a type of problems has been recently greatly popularized by the development of the field of nonadaptive compressed acquisition of data, the so-called *compressed sensing*, and its relationship with ℓ_1-norm minimization. We start our presentation by recalling the mathematical setting of compressed sensing as a reference framework for developing further generalizations. In particular we focus on the analysis of algorithms for such problems and their performances. We introduce and analyze the homotopy method, the iteratively re-weighted least squares method, and the iterative hard thresholding algorithm. We will see that the properties of convergence of these algorithms to the interesting solutions depend very much on special spectral properties, the Restricted Isometry Property or the Null Space Property, of the matrices which define the linear models. This provides a link to the analysis of random matrices which are typical examples of matrices with such properties. The concept of sparsity does not necessarily affect the entries of a vector only, but it can also be applied, for instance, to their variation. In the second part of the Chapter we address sparse optimizations in infinite dimensional spaces, and especially for situations where no Restricted Isometry Property or Null

Space Property are assumed for the linear model. We will be able to formulate efficient algorithms based on so-called iterative soft-thresholding also for such situations, although their analysis will require different tools, typically from nonsmooth convex analysis.
A common feature of the illustrated algorithms will be their variational nature, in the sense that they are derived as minimization strategies of given energy functionals. Not only does the variational framework allow us to derive very precise statements about the convergence properties of these algorithms, but it also provides the algorithms with an intrinsic robustness.
We conclude the Chapter with relatively new developments related to the negative contribution of noise to the recovery and the effectiveness of the algorithms described above, whether it is affecting the measurements or it is influencing the source before the measurements. As we discuss here the latter situation is unfortunately in the field of sparse recovery a limiting aspect as the noise gets amplified tremendously by the measurements on the noisy signal and completely new algorithms need to be developed to filter in a smart way such disturbance.

We made an effort to provide a concise overview of the most relevant algorithms in sparse recovery, hopefully in a rather simple way and at the same time to furnish enough mathematical insights to allow the formal comprehension of their behaviors. Additionally we provide extensive numerical examples and comparative tests of all the presented methods, which allow for more conscious choices in their practical use.

1.1 Notations

In the following we collect general notations. More specific notations may be introduced and recalled in the following sections.
We will consider often $\mathbb{R}^N$ as a Banach space endowed with different norms. In particular, later we use the ℓ_p-(quasi-)norms

$$\|x\|_p := \|x\|_{\ell_p} := \begin{cases} \left(\sum_{i=1}^N |x_j|^p\right)^{1/p}, \ 0 < p < \infty, \\ \max_{j=1,\dots,N} |x_j|, \ p = \infty. \end{cases} \tag{1}$$

Associated to these norms we denote their unit balls by $B_{\ell_p} := \{x \in \mathbb{R} : \|x\|_p \leq 1\}$ and the balls of radius R by $B_{\ell_p}(R) := R \cdot B_{\ell_p}$.
We will also consider infinite dimensional spaces. The index set $\mathcal{I}$ is supposed to be countable and we will consider the $\ell_p(\mathcal{I})$ spaces of p-summable sequences over the index set $\mathcal{I}$ as well. Their norm are defined as usual and similarly to the case of $\mathbb{R}^N$. We use the same notations B_{ℓ_p} for the $\ell_p(\mathcal{I})$-balls as for the ones in $\mathbb{R}^N$. With Φ we will usually denote a $m \times N$ real matrix, $m,N \in \mathbb{N}$ or an operator $\Phi : \ell_2(\mathcal{I}) \to Y$. We denote with Φ^* the adjoint of a matrix orthe adjoint of an operator. We will always work on real vector spaces, hence, in finite dimensions, Φ^* usually coincides with the transposed matrix of Φ. The norm of an operator $\Phi : X \to Y$ acting between two Banach spaces is denoted by $\|\Phi\|_{X \to Y}$; for matrices the norm $\|\Phi\|$ denotes the spectral norm. The set $\mathcal{F}_\Phi(y) := \{x \in X : \Phi x = y\}$ denotes the solution affine space in X of the linear system $\Phi x = y$ and $\mathcal{N}_\Phi$ the null space of a matrix or an operator Φ. The support of a vector $u \in \ell_2(\mathcal{I})$, i.e., the set of coordinates which are not zero, is denoted by $\operatorname{supp}(u) = \{i \in \mathcal{I} : u_i \neq 0\}$.
We will consider sub-index sets $\Lambda \subset \mathcal{I}$ and their complements $\Lambda^c = \mathcal{I} \setminus \Lambda$. The symbols $|\Lambda|$ and $\#\Lambda$ are used indifferently for indicating the cardinality of Λ. With a slight abuse we will denote

$$\|u\|_0 := \|u\|_{\ell_0(\mathcal{I})} := \#\mathrm{supp}(u), \tag{2}$$

which is popularly called the "ℓ_0-norm" in the literature. When $\#\mathcal{I} = N$ then $\ell_2(\mathcal{I}) = \mathbb{R}^N$ and we may also denote $\|u\|_{\ell_0} := \|u\|_{\ell_0(\mathcal{I})}$ We use the notation Φ_Λ to indicate a submatrix extracted from Φ by retaining only the columns indexed in Λ as well as the restrictions u_Λ of vectors u to the index set Λ. We also denote by $\Phi^*\Phi_{\Lambda\times\Lambda} := (\Phi^*\Phi)_{\Lambda\times\Lambda} := \Phi_\Lambda^*\Phi_\Lambda$ the submatrix extracted from $\Phi^*\Phi$ by retaining only the entries indexed on $\Lambda\times\Lambda$. In particular Φ_i is the i-th column of the matrix Φ. We denote the identity matrix by I.
Generic positive constants used in estimates are denoted as usual by

$$c, C, \tilde{c}, \tilde{C}, c_0, C_0, c_1, C_1, c_2, C_2, \ldots.$$

1.2 A Toy Mathematical Model for Sparse Recovery

1.2.1 Adaptive and Compressed Acquisition

Let $k, N \in \mathbb{N}$, $k \leq N$ and

$$\Sigma_k := \{x \in \mathbb{R}^N : \|x\|_{\ell_0} := \#\mathrm{supp}(x) \leq k\},$$

be the set of vectors with at most k nonzero entries, which we will call *k-sparse vectors*. The *k-best approximation error* that we can achieve in this set to a vector $x \in \mathbb{R}^N$ with respect to a suitable space quasi-norm $\|\cdot\|_X$ is defined by

$$\sigma_k(x)_X = \inf_{z\in\Sigma_k} \|x - z\|_X.$$

Example 1. Let $r(x)$ be the *nonincreasing rearrangement* of x, i.e.,

$$r(x) = (|x_{i_1}|, \ldots, |x_{i_N}|)^* \text{ and } |x_{i_j}| \geq |x_{i_{j+1}}|, \text{ for } j = 1, \ldots, N-1.$$

Then it is straightforward to check that

$$\sigma_k(x)_{\ell_p} := \left(\sum_{j=k+1}^{N} r_j(x)^p \right)^{1/p}, \quad 1 \leq p < \infty.$$

In particular, the vector $x_{[k]}$ derived from x by setting to zero all the $N-k$ smallest entries in absolute value is called the *best k-term approximation* to x and it coincides with

$$x_{[k]} = \arg\min_{z\in\Sigma_k} \|x - z\|_{\ell_p}. \tag{3}$$

for any $1 \leq p < \infty$.

The computation of the best k-term approximation of $x \in \mathbb{R}^N$ in general requires the search of the largest entries of x in absolute value, and therefore the testing of all the entries of the vector x. This procedure is *adaptive*, since it depends on the particular vector, and it is currently at the basis of lossy compression methods, such as JPEG [50]. Now that we understood what does it mean *adaptive compression* we may wonder whether we can compress a vector x by means of a *nonadaptive* linear testing.

1.2.2 Nonadaptive and Compressed Acquisition: Compressed Sensing

One would like to describe a *linear encoder* which allows us to compute approximately k measurements $(y_1,\dots,y_k)^*$ and a nearly optimal approximation of x in the following sense:

Provided a set $K \subset \mathbb{R}^N$, there exists a linear map $\Phi : \mathbb{R}^N \to \mathbb{R}^m$, with $m \approx k$ and a possibly nonlinear map $\Delta : \mathbb{R}^m \to \mathbb{R}^N$ such that

$$\|x - \Delta(\Phi x)\|_X \leq C\sigma_k(x)_X$$

for all $x \in K$.

Note that the way we encode $y = \Phi x$ is via a prescribed map Φ which is independent of x. Also the decoding procedure Δ might depend on Φ, but not on x. This is why we call this strategy a *nonadaptive (or universal) and compressed acquisition* of x. Note further that we would like to recover an approximation to x from nearly k-linear measurements which is of the order of the k-best approximation error. In this sense we say that the performances of the encoder/decoder system (Φ,Δ) is nearly optimal, because it matches (asymptotically, i.e., up to a multiplicative constant) the best possible adaptive compression error, modelled here by $\sigma_k(x)_X$.

1.3 Survey on Mathematical Analysis of Compressed Sensing

In the following sections we want to show that under a certain property, called the *Restricted Isometry Property* (RIP) for a matrix Φ, depending on the parameter $k \in \mathbb{N}$, $1 \leq k \leq m \leq N$, we have

The decoder, which we call ℓ_1*-minimization*,

$$\Delta(y) = \arg\min_{\Phi z = y = \Phi x} \|z\|_{\ell_1} \tag{4}$$

performs

$$\|x - \Delta(y)\|_{\ell_1} \leq C_1 \sigma_k(x)_{\ell_1}, \tag{5}$$

as well as

$$\|x - \Delta(y)\|_{\ell_2} \leq C_2 \frac{\sigma_k(x)_{\ell_1}}{k^{1/2}}, \tag{6}$$

for all $x \in \mathbb{R}^N$.

Hence this decoder is nearly optimal in the sense mentioned before. Additionally, we mention here that the best possible k for which (5) and (6) hold must necessarily satisfy

$$k \leq C \frac{m}{\log N/m + 1},$$

for some constant C independent of m and N and we mention below in Section 1.3.4 that certain *randomly generated* matrices Φ do possess the RIP with k precisely scaling like that with high probability. However,

we do not enter in these details on the complexity of the nonadaptive acquision problem, and we may refer to [33] for more insights.

1.3.1 An Intuition Why ℓ_1-Minimization Works Well

In this section we would like to provide an intuitive explanation of the near-optimal error estimates (5) and (6) provided by ℓ_1-minimization (4) in recovering vectors from partial linear measurements. Equations (5) and (6) ensure in particular that if the vector x is *k-sparse*, then ℓ_1-minimization (4) will be able to recover it *exactly* from m linear measurements y obtained via the matrix Φ. In fact for a k-sparse vector $x = x_{[k]}$ and $\sigma_k(x) = 0$. This result is quite surprising because the problem of recovering a sparse vector, or the solution of the following optimization

$$\min_{z \in \mathcal{F}_\Phi(y)} \|z\|_{\ell_0}, \tag{7}$$

is known to be *NP-complete*[1] [51, 52] whereas ℓ_1-minimization is a convex problem which can be solved at any prescribed accuracy in polynomial time. For instance interior-point methods are guaranteed to solve the ℓ_1-problem to a fixed precision in time $\mathcal{O}(m^2N^{1.5})$ [55]. The first intuitive approach to this perhaps surprising result is by interpreting ℓ_1-minimization as the *convex relaxation* of the problem (7).

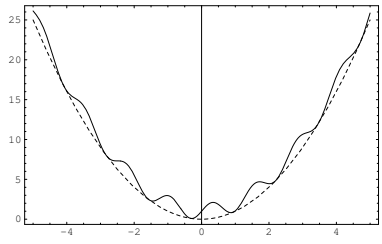


Fig. 1 A nonconvex function f and a convex approximation $g \leq f$ from below.

If we were interested in solving an optimization problem

$$\min_{z \in \mathcal{C}} f(z),$$

where f is a nonconvex, lower-semincontinuous function, and $\mathcal{C}$ is a closed convex set, it might be convenient to recast the problem by considering its convexification, i.e.,

$$\min_{z \in \mathcal{C}} \bar{f}(z),$$

where $\bar{f}$ is called the *convex relaxation* or the *convex envelope* of f and is given by

$$\bar{f}(x) := \sup\{g(x) \leq f(x) : g \text{ is a convex function}\}.$$

[1] NP stands for non-deterministic polynomial-time and indicates a class of problems for which the verification of their solution has a computational costs which is polynomial in the size of the input. However presently it is not known whether such problems can be solved with a polynomial complexity algorithm. This issue is the first in the list of the *Millennium Prize Problems* of the Clay Mathematics Institute.

The motivation of this choice is simply geometrical. While f can have many minimizers on $\mathcal{C}$, its convex envelop $\bar{f}$ has global minimizers, and such global minimizers are likely to be in a neighborhood of a global minimizer of f, see Figure 1. Actually if $\mathcal{C}$ is compact, then the global minima of f and $\bar{f}$ must necessarily coincide. Unfortunately, the precise computation of $\bar{f}$ is again a very difficult problem. In the case of $\|x\|_{\ell_0}$,

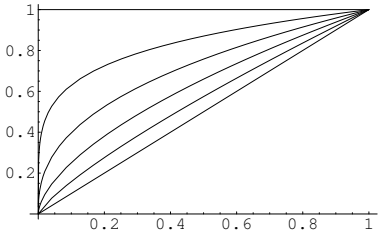


Fig. 2 The absolute value function $|\cdot|$ is the convex relaxation of the function $|\cdot|_0$ on $[0,1]$.

one rewrites

$$\|x\|_{\ell_0} := \sum_{j=1}^{N} |x_j|_0, \quad |t|_0 := \begin{cases} 0, & t = 0 \\ 1, & t \neq 0 \end{cases}.$$

Its convex envelope in $B_{\ell_\infty}(R) \cap \mathcal{F}_\Phi(y)$ is bounded below by $\frac{1}{R}\|x\|_{\ell_1} := \frac{1}{R}\sum_{j=1}^{N} |x_j|$, see Figure 2. This observation gives already a first impression of the motivation why ℓ_1-minimization can help in approximating sparse solutions of $\Phi x = y$. However, it is not yet clear when a global minimizer of the ℓ_1-minimization (4) really coincides with a solution to (7), since the ℓ_1-norm is not yet the precise convex envelope of $\|\cdot\|_{\ell_0}$ over the solution space $\mathcal{F}_\Phi(y)$. Again a simple geometrical reasoning can help us to get a feeling about more general principles which will be addressed more formally in the following sections.

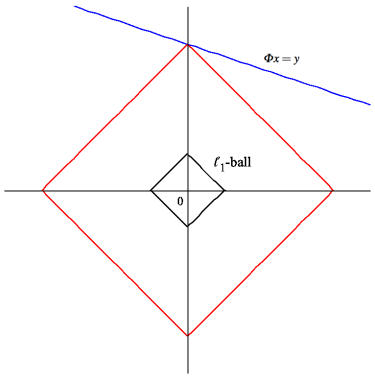


Fig. 3 The ℓ_1-minimizer within the affine space of solutions of the linear system $\Phi x = y$ coincides with the sparsest solution.

Assume for a moment that $N = 2$ and $m = 1$. Hence we are dealing with an affine space of solutions $\mathcal{F}_\Phi(y)$ which is just a line in $\mathbb{R}^2$. When we search for the ℓ_1- norm minimizers among the elements $\mathcal{F}_\Phi(y)$ (see Figure 3), we immediately realize that, except for pathological situations where $\mathcal{N}_\Phi = \ker \Phi$ is parallel to one of the faces of the polytope B_{ℓ_1}, there is a unique solution which coincides also with a solution with a

minimal number of nonzero entries. Therefore, if we exclude situations in which there exists $\eta \in \mathcal{N}_\Phi$ such that $|\eta_1| = |\eta_2|$ or, equivalently, we assume that

$$|\eta_i| < |\eta_{\{1,2\}\setminus\{i\}}| \tag{8}$$

for all $\eta \in \mathcal{N}_\Phi$ and for one $i = 1,2$, then the solution to (4) is a solution to (7)! Note also that, if we give a uniform probability distribution to the angle in $[0,2\pi]$ formed by $\mathcal{N}_\Phi$ and any of the coordinate axes, then we realize that the pathological situation of violating (8) has zero probability. Of course, in higher dimension such simple reasoning becomes more involved, since the number of faces and edges of an ℓ_1-ball B_{ℓ_1} becomes larger and larger and one should cumulate the probabilities of different angles with respect to possible affine spaces of codimension $N-m$. However, condition (8) is the right prototype of a property (we call it the *Null Space Property* (NSP) and we describe it in detail in the next section) which guarantees, also in higher dimension, that the solution to (4) is a solution to (7).

1.3.2 Restricted Isometry Property and Null Space Property

Definition 1. One says that $\Phi \in \mathbb{R}^{m\times N}$ has the *Null Space Property* (NSP) of order k, $1 \le k \le m$, for $0 < \gamma_k < 1$ (and we may write that Φ has the (k,γ_k)-NSP) if

$$\|\eta_\Lambda\|_{\ell_1} \le \gamma_k \|\eta_{\Lambda^c}\|_{\ell_1}, \tag{9}$$

for all sets $\Lambda \subset \{1,\dots,N\}$, $\#\Lambda \le k$ and for all $\eta \in \mathcal{N}_\Phi = \ker\Phi$.

Note that this definition essentially generalizes condition (8) which we introduced by our simple and rough geometrical reasoning in $\mathbb{R}^2$. Notice that the NSP essentially excludes that in the null space of Φ one can find sparse vectors (otherwise they would be indistinguishable from 0), but also vectors with few very large components in absolute value with respect to all the others. Further we need a related property for matrices.

Definition 2. One says that $\Phi \in \mathbb{R}^{m\times N}$ has the *Restricted Isometry Property* (RIP) of order K if there exists $0 < \delta_K < 1$ such that

$$(1-\delta_K)\|z\|_{\ell_2} \le \|\Phi z\|_{\ell_2} \le (1+\delta_K)\|z\|_{\ell_2},$$

for all $z \in \Sigma_K$. In this case we may write that Φ has the (K,δ_K)-RIP.

Notice that this propery is also excluding that sparse vectors are elements of the kernel of Φ, otherwise we would have $0 = \|\Phi z\|_{\ell_2} \ge (1-\delta_K)\|z\|_{\ell_2} \ne 0$ for $z \in \mathcal{N}_\Phi \setminus \{0\} \cap \Sigma_k$.

The RIP turns out to be very useful in the analysis of stability of certain algorithms as we will show in Section 2.3. The RIP is also introduced because it implies the *Null Space Property*, and when dealing with random matrices (see Section 1.3.4) it is more easily addressed. In fact we have:

Lemma 1. *Assume that $\Phi \in \mathbb{R}^{m\times N}$ has the RIP of order $K = k+h$ with $0 < \delta_K < 1$. Then Φ has the NSP of order k and constant $\gamma_k = \sqrt{\frac{k}{h}}\frac{1+\delta_K}{1-\delta_K}$.*

Proof. Let $\Lambda \subset \{1,\dots,N\}$, $\#\Lambda \le k$. Define $\Lambda_0 = \Lambda$ and $\Lambda_1, \Lambda_2, \dots, \Lambda_s$ disjoint sets of indexes of size at most h, associated to a decreasing rearrangement of the entries of $\eta \in \mathcal{N}_\Phi = \ker(\Phi)$. Then, by using Cauchy-Schwarz inequality, the RIP twice, the fact that $\Phi\eta = 0$, and eventually the triangle inequality, we have the following sequence of inequalities:

$$\|\eta_\Lambda\|_{\ell_1} \le \sqrt{k}\|\eta_\Lambda\|_{\ell_2} \le \sqrt{k}\|\eta_{\Lambda_0\cup\Lambda_1}\|_{\ell_2} \le (1-\delta_K)^{-1}\sqrt{k}\|\Phi\eta_{\Lambda_0\cup\Lambda_1}\|_{\ell_2} = (1-\delta_K)^{-1}\sqrt{k}\|\Phi\eta_{\Lambda_2\cup\Lambda_3\cup\cdots\cup\Lambda_s}\|_{\ell_2}$$
$$\le (1-\delta_K)^{-1}\sqrt{k}\sum_{j=2}^{s}\|\Phi\eta_{\Lambda_j}\|_{\ell_2} \le \frac{1+\delta_K}{1-\delta_K}\sqrt{k}\sum_{j=2}^{s}\|\eta_{\Lambda_j}\|_{\ell_2}. \tag{10}$$

Note now that $i \in \Lambda_{j+1}$ and $\ell \in \Lambda_j$ imply by construction of $\Lambda_j's$ by nonincreasing rearrangement of the entries of η

$$|\eta_i| \le |\eta_\ell|.$$

By taking the sum over ℓ first and than the ℓ_2-norm over i we get

$$|\eta_i| \le h^{-1}\|\eta_{\Lambda_j}\|_{\ell_1}, \text{ and } \|\eta_{\Lambda_{j+1}}\|_{\ell_2} \le h^{-1/2}\|\eta_{\Lambda_j}\|_{\ell_1}.$$

By using the latter estimates in (10) we obtain

$$\|\eta_\Lambda\|_{\ell_1} \le \frac{1+\delta_K}{1-\delta_K}\sqrt{\frac{k}{h}}\sum_{j=1}^{s-1}\|\eta_{\Lambda_j}\|_{\ell_1} \le \left(\frac{1+\delta_K}{1-\delta_K}\sqrt{\frac{k}{h}}\right)\|\eta_{\Lambda^c}\|_{\ell_1}. \quad \square$$

The RIP property does imply the NSP, but the converse is not true. Actually the RIP is significantly more restrictive.

1.3.3 More on RIP

In the following, we introduce some further useful technical lemmas which shed light on fundamental properties of RIP matrices and sparse approximations, as established in [71].

Lemma 2. *For all index sets $\Lambda \subset \{1,\dots,N\}$ and all Φ for which the RIP holds with order $k = |\Lambda|$, we have*

$$\|\Phi_\Lambda^* y\|_{\ell_2} \le (1+\delta_k)\|y\|_{\ell_2}, \tag{11}$$

$$(1-\delta_k)^2\|x_\Lambda\|_{\ell_2} \le \|\Phi_\Lambda^*\Phi_\Lambda x_\Lambda\|_{\ell_2} \le (1+\delta_k)^2\|x_\Lambda\|_{\ell_2}, \tag{12}$$

and

$$\|(I-\Phi_\Lambda^*\Phi_\Lambda)x_\Lambda\|_{\ell_2} \le 3\delta_k\|x_\Lambda\|_{\ell_2}, \tag{13}$$

for arbitrary $x \in \mathbb{R}^N$ and $y \in \mathbb{R}^m$. Furthermore, for two disjoint sets Λ_1 and Λ_2 and all A for which the RIP holds with order $k = |\Lambda|$, $\Lambda = \Lambda_1 \cup \Lambda_2$,

$$\|\Phi_{\Lambda_1}^*\Phi_{\Lambda_2}x_{\Lambda_2}\|_{\ell_2} \le 3\delta_k\|x_\Lambda\|_{\ell_2}. \tag{14}$$

Proof. To see (11), note that $\left\|\Phi_\Lambda^* y\right\|_{\ell_2}^2 = \left\langle \Phi_\Lambda^* y, \Phi_\Lambda^* y\right\rangle = \left\langle \Phi_\Lambda\Phi_\Lambda^* y, y\right\rangle$. The assertion now follows by an application of the Cauchy-Schwarz inequality and the RIP. The right-hand inequality of (12) is an application of 11 for $y := \Phi_\Lambda x_\Lambda$ and again the RIP. The left-hand inequality again follows by an application of the Cauchy-Schwarz inequality, i.e., $\|x_\Lambda\|_{\ell_2}\left\|\Phi_\Lambda^*\Phi_\Lambda x_\Lambda\right\|_{\ell_2} \ge \left\langle x_\Lambda, \Phi_\Lambda^*\Phi_\Lambda x_\Lambda\right\rangle = \|\Phi_\Lambda x_\Lambda\|_{\ell_2}^2$, and again an application of the RIP. The proof of (13) follows from the fact that the spectrum of the matrix $\Phi_\Lambda^*\Phi_\Lambda$ is bounded contained in $[(1-\delta_k)^2,(1+\delta_k)^2]$ due to (12). Thus, one obtains the assertion by means of $\|(I-\Phi_\Lambda^*\Phi_\Lambda)\| \le \max\{(1+\delta_k)^2-1, 1-(1-\delta_k)^2\} = \delta_k(2+\delta_k) \le 3\delta_k$. For (14), just note that $\Phi_{\Lambda_1}^*\Phi_{\Lambda_2}$ is a

submatrix of $\Phi^*_{\Lambda_1\cup\Lambda_2}\Phi_{\Lambda_1\cup\Lambda_2}-I$, and therefore $\|\Phi^*_{\Lambda_1}\Phi_{\Lambda_2}\|\le\|I-\Phi^*_{\Lambda_1\cup\Lambda_2}\Phi_{\Lambda_1\cup\Lambda_2}\|$. One concludes by (13). □

Lemma 3. *Suppose the matrix Φ satisfies the RIP of order k with constant $\delta_k>0$. Then for all vectors x, the following bound holds*

$$\|\Phi x\|_{\ell_2}\le(1+\delta_k)\left(\|x\|_{\ell_2}+\frac{\|x\|_{\ell_1}}{k^{1/2}}\right). \tag{15}$$

Proof. For a proof of this Lemma we refer to the one of [71, Proposition 3.5]. □

1.3.4 Random Matrices and Optimal RIP

In this section we would like to mention exemplarily for Gaussian and Bernoulli random matrices how it is possible to show that the RIP property can hold with optimal constants, i.e.,

$$k\asymp\frac{m}{\log N/m+1}.$$

at least with high probability. This implies in particular, that such matrices exist, they are frequent, but they are given to us only with an uncertainty.

Let $(\Omega,\mathbb{P})$ be a probability space and $\mathcal{X}$ a random variable on $(\Omega,\mathbb{P})$. One can define a random matrix $\Phi(\omega)$, $\omega\in\Omega^{m\times N}$, as the matrix whose entries are independent realizations of $\mathcal{X}$. We assume further that $\|\Phi(\omega)x\|^2_{\ell_2}$ has expected value $\|x\|^2_{\ell_2}$ and

$$\mathbb{P}\left(\left|\|\Phi(\omega)x\|^2_{\ell_2}-\|x\|^2_{\ell_2}\right|\ge\varepsilon\|x\|^2_{\ell_2}\right)\le 2e^{-mc_0(\varepsilon)},\quad 0<\varepsilon<1. \tag{16}$$

Example 2. Here we collect two relevant examples for which the concentration property (16) holds:

1. One can choose, for instance, the entries of Φ as i.i.d. Gaussian random variables, $\Phi_{ij}\sim\mathcal{N}(0,\frac{1}{m})$, and $c_0(\varepsilon)=\varepsilon^2/4-\varepsilon^3/6$. This can be shown by using Chernoff inequalities and a comparison of the moments of a Bernoulli random variable with respect to those of a Gaussian random variable;
2. One can also use matrices where the entries are independent realizations of ±1 Bernoulli random variables, i.e.,
$$\Phi_{ij}=\begin{cases}+1/\sqrt{m}, & \text{with probability } \frac{1}{2}\\ -1/\sqrt{m}, & \text{with probability } \frac{1}{2}\end{cases}.$$

Then we have the following result, as shown in [4]. It follows by an application of the concentration inequality (16) on a discrete and finite set of sparse vectors approximating any sparse vector of unit norm up to a small distortion and then by applying a continuity argument to extend the validity of the concentration inquality to all sparse vectors with the same probability.

Theorem 1. *Suppose that m, N and $0<\delta<1$ are fixed. If $\Phi(\omega),\omega\in\Omega^{mN}$ is a random matrix of size $m\times N$ with the concentration property* (16)*, then there exist constants $c_1,c_2>0$ depending on δ such that the RIP holds for $\Phi(\omega)$ with constant δ and $k\le c_1\frac{m}{\log(N/m)+1}$ with probability exceeding $1-2e^{-c_2m}$.*

An extensive study on RIP properties of different types of matrices, for instance partial orthogonal matrices or random structured matrices, such as partial Fourier matrices or partial circulant matrices, is provided in [33, 65].

1.3.5 Performances of ℓ_1-Minimization as an Optimal Decoder

In this section we address the proofs of the approximation properties (5) and (6). As they are elementary and very instructive, we perform them in full detail.

Theorem 2. *Let $\Phi \in \mathbb{R}^{m\times N}$ satisfy the RIP of order $2k$ with $\delta_{2k} \le \delta < \frac{\sqrt{2}-1}{\sqrt{2}+1}$ (or simply Φ satisfies the NSP of order k with constant $\gamma_k = \frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}} < 1$; to show this implication use the same arguments of the proof of Lemma 1 for $\Lambda_0 = \Lambda$, $\#\Lambda_1 = k$, while $\#\Lambda_j \le 2k$ for $j \ge 2$) , then the decoder Δ as in* (4) *satisfies* (5)*.*

Proof. By using the same arguments of Lemma 1, but for $\Lambda_0 = \Lambda$, $\#\Lambda_1 = k$, while $\#\Lambda_j \le 2k$ for $j \ge 2$, we have

$$\|\eta_\Lambda\|_{\ell_1} \le \frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}}\|\eta_{\Lambda^c}\|_{\ell_1},$$

for all $\Lambda \subset \{1,\dots,N\}$, $\#\Lambda \le k$ and $\eta \in \mathcal{N}_\Phi = \ker\Phi$. Let $x^* = \Delta(\Phi x)$, so that $\eta = x^* - x \in \mathcal{N}_\Phi$, and

$$\|x^*\|_{\ell_1} \le \|x\|_{\ell_1}.$$

One denotes now with Λ the set of the k-largest entries of x in absolute value. One has

$$\|x^*_\Lambda\|_{\ell_1} + \|x^*_{\Lambda^c}\|_{\ell_1} \le \|x_\Lambda\|_{\ell_1} + \|x_{\Lambda^c}\|_{\ell_1}.$$

It follows immediately by triangle inequality

$$\|x_\Lambda\|_{\ell_1} - \|\eta_\Lambda\|_{\ell_1} + \|\eta_{\Lambda^c}\|_{\ell_1} - \|x_{\Lambda^c}\|_{\ell_1} \le \|x_\Lambda\|_{\ell_1} + \|x_{\Lambda^c}\|_{\ell_1}.$$

Hence

$$\|\eta_{\Lambda^c}\|_{\ell_1} \le \|\eta_\Lambda\|_{\ell_1} + 2\|x_{\Lambda^c}\|_{\ell_1} \le \frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}}\|\eta_{\Lambda^c}\|_{\ell_1} + 2\sigma_k(x)_{\ell_1},$$

or, equivalently,

$$\|\eta_{\Lambda^c}\|_{\ell_1} \le \frac{2}{1-\frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}}}\sigma_k(x)_{\ell_1}. \tag{17}$$

In particular, note that for $\delta < \frac{\sqrt{2}-1}{\sqrt{2}+1}$ we have $\frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}} < 1$. Eventually we conclude the estimates

$$\|x - x^*\|_{\ell_1} = \|\eta_\Lambda\|_{\ell_1} + \|\eta_{\Lambda^c}\|_{\ell_1} \le \left(\frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}} + 1\right)\|\eta_{\Lambda^c}\|_{\ell_1} \le C_1\sigma_k(x)_{\ell_1},$$

where $C_1 := \left\lceil \frac{2\left(\frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}}+1\right)}{1-\frac{1+\delta}{1-\delta}\sqrt{\frac{1}{2}}} \right\rceil$. □

Similarly we address the second estimate (6).

Theorem 3. *Let $\Phi \in \mathbb{R}^{m\times N}$ satisfy the RIP of order $3k$ with $\delta_{3k} \le \delta < \frac{\sqrt{2}-1}{\sqrt{2}+1}$, then the decoder Δ as in* (4) *satisfies* (6)*.*

Proof. Let $x^* = \Delta(\Phi x)$. As we proceeded in Lemma 1, we denote $\eta = x^* - x \in \mathcal{N}_\Phi$, $\Lambda_0 = \Lambda$ the set of the $2k$-largest entries of η in absolute value, and Λ_j of size at most k composed of nonincreasing rearrangement entries of η. Then

$$\|\eta_\Lambda\|_{\ell_2} \leq \frac{1+\delta}{1-\delta} k^{-\frac{1}{2}} \|\eta_{\Lambda^c}\|_{\ell_1}.$$

Note that $\|\eta_{\Lambda^c}\|_{\ell_2} \leq (2k)^{-\frac{1}{2}} \|\eta\|_{\ell_1}$ (see, e.g., [31, Lemma 2.2] for a proof). By means of this result and by Lemma 1

$$\|\eta_{\Lambda^c}\|_{\ell_2} \leq \frac{1}{(2k)^{\frac{1}{2}}} \|\eta\|_{\ell_1} = \frac{1}{(2k)^{\frac{1}{2}}} \left(\|\eta_\Lambda\|_{\ell_1} + \|\eta_{\Lambda^c}\|_{\ell_1}\right) \leq \frac{1}{(2k)^{\frac{1}{2}}} \left(C\|\eta_{\Lambda^c}\|_{\ell_1} + \|\eta_{\Lambda^c}\|_{\ell_1}\right) = \frac{C+1}{\sqrt{2}} k^{-\frac{1}{2}} \|\eta_{\Lambda^c}\|_{\ell_1},$$

for a suitable constant $C > 0$. Note that, being Λ the set of the in absolute value $2k$-largest entries of η, one has also

$$\|\eta_{\Lambda^c}\|_{\ell_1} \leq \|\eta_{(\mathrm{supp}\, x_{[2k]})^c}\|_{\ell_1} \leq \|\eta_{(\mathrm{supp}\, x_{[k]})^c}\|_{\ell_1}, \tag{18}$$

where $x_{[h]}$ is the best h-term approximation to x. The use of this latter estimate, combined with inequality (17) finally gives

$$\|x - x^*\|_{\ell_2} \leq \|\eta_\Lambda\|_{\ell_2} + \|\eta_{\Lambda^c}\|_{\ell_2} \leq C_1 k^{-1/2} \|\eta_{\Lambda^c}\|_{\ell_1} \leq C_2 k^{-1/2} \sigma_k(x)_{\ell_1}.$$

□

1.4 Noise Models

So far we were only considering the very simple model

$$y = \Phi x, \tag{19}$$

where the measurement $y \in \mathbb{R}^m$ is obtained by a simple linear acquisition not affected by any disturbance.

1.4.1 Measurement Noise and Model Error

In general, real-life applications are not as exact and data are affected by disturbances and one typically considers model problems of the type

$$y = \Phi x + e, \tag{20}$$

where an additional deterministic or random noise vector $e \in \mathbb{R}^m$ corrupts the linear measurements. It is also legitimate to interpret e as a model error, which may be due to the fact that the measurement process is only approximately linear. Regarding this modified model, an enhanced stability property of ℓ_1-minimization is established in [10]. In the following we state this result, preceded by an auxiliary lemma.

Lemma 4. *For any x we denote $x^{[k]} = x - x_{[k]}$, where $x_{[k]}$ is the best k-term approximation to x. Let*

$$y = \Phi x + e = \Phi x_{[k]} + \Phi x^{[k]} + e = \Phi x_{[k]} + \tilde{e}.$$

If Φ has the RIP of order k and constant δ_k, then the norm of the error $\tilde{e}$ can be bounded by

$$\|\tilde{e}\|_{\ell_2} \leq (1+\delta_k)\left(\sigma_k(x)_{\ell_2} + \frac{\sigma_k(x)_{\ell_1}}{k^{1/2}}\right) + \|e\|_{\ell_2}. \tag{21}$$

Proof. Decompose $x = x_{[k]} + x^{[k]}$ and $\tilde{e} = \Phi x^{[k]} + e$. To compute the norm of the error term, we simply apply the triangle inequality and Lemma 3. □

Theorem 4. *Let $\Phi \in \mathbb{R}^{m\times N}$ which satisfies the RIP of order $2k$ with constant δ_{2k} sufficiently small. Assume further that $y = \Phi x + e$ where e is a measurement error, and that $x^* = \Delta(y)$ is k-sparse. Then the decoder Δ has the further enhanced stability property:*

$$\|x - \Delta(y)\|_{\ell_2} \leq C_3\left(\sigma_k(x)_{\ell_2} + \frac{\sigma_k(x)_{\ell_1}}{k^{1/2}} + \|e\|_{\ell_2}\right). \tag{22}$$

Proof. Let us consider the following estimates, which follows by an application of the lower bound of the RIP and Lemma 4:

$$\begin{aligned}
\|x - \Delta(y)\|_{\ell_2} &\leq \|x^{[k]}\|_{\ell_2} + \|x_{[k]} - x^*\|_{\ell_2} \\
&\leq \sigma_k(x)_{\ell_2} + \frac{1}{1-\delta_{2k}}\|\Phi x_{[k]} - \Phi x^*\|_{\ell_2} \\
&= \sigma_k(x)_{\ell_2} + \frac{1}{1-\delta_{2k}}\|\Phi x_{[k]} - y\|_{\ell_2} \\
&= \sigma_k(x)_{\ell_2} + \frac{1}{1-\delta_{2k}}\|\tilde{e}\|_{\ell_2} \\
&\leq C_3\left(\sigma_k(x)_{\ell_2} + \frac{\sigma_k(x)_{\ell_1}}{k^{1/2}} + \|e\|_{\ell_2}\right).
\end{aligned}$$

□

For some types of random matrices (for instance Gaussian random matrices) the stability estimate (22) holds with high probability without assuming the sparsity of x^*. In genera other measurement matrices will not perform as good and the ℓ_1-minimization decoder Δ is maybe not the most suitable decoder when noise is present on the data. We refer to the so-called "quotient property" as analyzed in [33, Chapter 11].
There is a vast literature, which considers this problem and a bunch of alternative decoders to ℓ_1 minimization were proposed. The most intuitive approach to cover noisy measurements is the *Basis Pursuit Denoising* (BPDN) problem

$$\Delta_{\mathrm{DN}}(y) = \arg\min_{\|\Phi z - y\|_{\ell_2}\leq\eta} \|z\|_{\ell_1}, \tag{23}$$

where one has to tune the parameter $0 < \eta \approx \|e\|_{\ell_2}$. Let us mention that in the literature the standard ℓ_1-minization decoding process Δ is also referred to as *Basis Pursuit* (BP). Another approach is the so called *Least Absolute Shrinkage and Selection Operator* (LASSO) [69]

$$\Delta_{\mathrm{LA}}(y) = \arg\min_{\|z\|_{\ell_1}\leq\eta} \|\Phi z - y\|_{\ell_2}, \tag{24}$$

for a positive parameter η. It can be seen as a convexification of the more intuitive problem formulation

$$\arg\min_{\|z\|_{\ell_0}\leq k} \|\Phi z - y\|_{\ell_2}, \tag{25}$$

where one wishes to restrict the sparsity of the signal to be recovered. A popular formulation is the ℓ_1-*regularized least squares problem*

$$\Delta_\lambda(y) = \arg\min\left(\mathcal{J}_\lambda(z) := \lambda\|z\|_{\ell_1} + \frac{1}{2}\|\Phi z - y\|_{\ell_2}^2\right), \tag{26}$$

which again can be considered as a convexification of the ℓ_0-*regularized least squares problem*

$$\Delta_{0,\lambda}(y) = \arg\min\left(\mathcal{J}_0(z) := \lambda\|z\|_{\ell_0} + \frac{1}{2}\|\Phi z - y\|_{\ell_2}^2\right), \tag{27}$$

where the regularization parameter λ controls the balance of the fidelity and penalty term. Problems (23), (24), and (26) are equivalent for suitable values of ε, η, and λ, thus, in the literature, also problem (26) is often referred to as LASSO or BPDN. Define the minimizers $x_\lambda := \arg\min_z \mathcal{J}_\lambda(z)$. When $\lambda = \hat{\lambda}$ is large enough then $x_{\hat{\lambda}} = 0$, and furthermore, $\lim_{\lambda\to 0} x_\lambda = x^*$, where $x^* = \Delta(y)$, i.e., the solution to (4).

The minimizer of $\mathcal{J}_\lambda$ can be characterized using the *subdifferential* [26], which is defined for a general convex function $F : \mathbb{R}^N \to \mathbb{R}$ at a point $x \in \mathbb{R}^N$ by

$$\partial F(x) = \{v \in \mathbb{R}^N, F(y) - F(x) \geq \langle v, y - x\rangle \text{ for all } y \in \mathbb{R}^N\}.$$

Clearly, x is a minimizer of F if and only if $0 \in \partial F(x)$. The subdifferential of $\mathcal{J}_\lambda$ is given by

$$\partial\mathcal{J}_\lambda(x) = \Phi^*(\Phi x - y) + \lambda\partial\|\cdot\|_{\ell_1}(x),$$

where the subdifferential of the ℓ_1-norm is given by

$$\partial\|\cdot\|_{\ell_1}(x) = \{v \in \mathbb{R}^N : v_\ell \in \partial|\cdot|(x_\ell), \ell = 1,\dots,N\},$$

with the subdifferential of the absolute value being

$$\partial|\cdot|(z) = \begin{cases} \{\operatorname{sign}(z)\} & \text{if } z \neq 0, \\ [-1,1] & \text{if } z = 0. \end{cases}$$

The inclusion $0 \in \partial\mathcal{J}_\lambda(x)$ is equivalent to

$$-(\Phi^*(\Phi x - y))_\ell = \lambda\operatorname{sign}(x_\ell) \qquad \text{if } x_\ell \neq 0, \tag{28}$$

$$|(\Phi^*(\Phi x - y))_\ell| \leq \lambda \qquad \text{if } x_\ell = 0, \tag{29}$$

for all $\ell = 1,\dots,N$. We use this characterization in Section 2.1 and Section 2.2.2.

1.4.2 Signal Noise and Noise Folding

In the previous section, we studied a model which takes corrupted measurements into account. However, in practice it is very uncommon to have the signal x detected by a certain device, totally free from some external noise. Therefore, instead of (20), it is reasonable to consider the more realistic model

$$y = \Phi(\bar{x}+n)+e, \tag{30}$$

where $\bar{x} \in \mathbb{R}^N$ is the noiseless signal and $n \in \mathbb{R}^N$ is the noise on the original signal. Trivially, by defining $\tilde{e} := \Phi n + e$, this model is reduced to $y = \Phi\bar{x} + \tilde{e}$, and thus to the situation in the preceding Section 1.4.1. However, the recent work [12, 70] shows how the measurement process actually causes the *noise-folding phenomenon*, which implies that the variance of the noise on the original signal is amplified by a factor of $\frac{N}{m}$, additionally contributing to the measurement noise, playing to our disadvantage in the recovery phase. Thus, the signal noise n seems to be much worse than the measurement noise e. More formally, if we add to the signal $\bar{x}$ a noise vector n, whose entries have normal distribution $\mathcal{N}(0,\sigma_n)$, the measurement y given by

$$y = \Phi(\bar{x}+n), \tag{31}$$

can be considered equivalently obtained by a measurement procedure of the form (20) (possibly with another measurement matrix Φ of equal statistics) where now the vector e is composed by i.i.d. Gaussian entries with distribution $\mathcal{N}(0,\sigma_e)$ and $\sigma_e^2 = \frac{N}{m}\sigma_n^2$.

Section 5 is entirely dedicated to the noise-folding phenomenon. We present two decoding procedures, combining ℓ_1-minimization followed by either a regularized selective least p-powers or an iterative hard thresholding, which not only from the beginning take the noise on the signal into account, but also have enhanced properties in terms of support identification with respect to the sole ℓ_1-minimization, BPDN, or iteratively re-weighted ℓ_1-minimization. We prove such features, providing relatively simple and precise theoretical guarantees. We additionally confirm and support the theoretical results by numerical simulations, which give a statistics of the robustness of the new decoding procedures with respect to more classical methods based on ℓ_1-minimization.

2 Numerical Methods for Compressed Sensing

The previous sections showed that ℓ_1-minimization performs very well in recovering sparse or approximately sparse vectors from undersampled measurements. In applications it is important to have fast methods for actually solving ℓ_1-minimization or at least with similar guarantees of stability. Three such methods – the homotopy (LARS) method introduced in [25, 61], the iteratively re-weighted least squares method (IRLS) [22], and the iterative hard thresholding algorithm [6, 7] – will be explained in more detail below.

As a first remark, the ℓ_1-minimization problem (4) is, in the case of real models Φ and data y, equivalent to the linear program

$$\min \sum_{j=1}^{N} v_j \quad \text{subject to} \quad v \geq 0,\ (\Phi| - \Phi)v = y. \tag{32}$$

The solution x^* to (4) is obtained from the solution v^* of (32) via $x^* = (I| - I)v^*$, for I the identity matrix. Any linear programming method may therefore be used for solving (4). The simplex method as well as interior point methods apply in particular [55], and standard software may be used. (In the complex case, (4) is equivalent to a second order cone program (SOCP) and can be solved by means of interior point methods as well.) However, such methods and software are of general purpose and one may expect that methods specialized to (4) outperform such existing standard methods. Moreover, standard software often has the drawback that one has to provide the full matrix rather than fast routines for matrix-vector multiplication which are available for instance in the case of partial Fourier matrices. In order to obtain the full performance of such methods one would therefore need to re-implement them, which is a daunting task because interior point

methods usually require much fine tuning. On the contrary the three specialized methods described below are rather simple to implement and very efficient. Many more methods are available nowadays, including greedy methods, such as Orthogonal Matching Pursuit [72] and CoSaMP [71]. However, only the three methods below are explained in detail because they highlight the fundamental concepts which are useful to comprehend also other algorithms.

2.1 The Homotopy Method (Modified LARS)

The homotopy method – or modified LARS – [24, 25, 59, 61] solves (4) and is a direct method, i.e., it solves the problem exactly in a finite number of steps.

One considers the ℓ_1-regularized least squares functionals $\mathcal{J}_\lambda$ of (26) and their minimizers x_λ, which are characterized by the conditions (28) and (29). Since $x_{\hat\lambda} = 0$ for $\lambda = \hat\lambda$ large enough, and $\lim_{\lambda\to 0} x_\lambda = x^*$, where x^* is the ℓ_1-minimizer, the idea of the homotopy method is to trace the solution x_λ from $x_{\hat\lambda} = 0$ to x^*. The crucial observation is that the solution path $\lambda \mapsto x_\lambda$ is piecewise linear, and it is enough to trace the endpoints of the linear pieces.

Thus, we define the starting point of the homotopy method $x^{(0)} = x_{\hat\lambda} = 0$. By conditions (28) and (29) the corresponding λ can be chosen as $\hat\lambda = \lambda^{(0)} = \|\Phi^* y\|_\infty$. In the further steps $j = 1, 2, \dots$ the algorithm computes minimizers $x^{(1)}, x^{(2)}, \dots$ and maintains an active (support) set Λ_j. Denote by

$$c^{(j)} = \Phi^*(y - \Phi x^{(j-1)})$$

the current residual vector. The columns of the matrix Φ are denoted by a_ℓ, $\ell = 1, \dots, N$ and for a subset $\Lambda \subset \{1, \dots, N\}$ we let Φ_Λ be the submatrix of Φ corresponding to the columns indexed by Λ.

Step 1: Let

$$\ell^{(1)} := \arg\max_{\ell=1,\dots,N} |(\Phi^* y)_\ell| = \arg\max_{\ell=1,\dots,N} |c_\ell^{(1)}|. \tag{33}$$

One assumes here and also in the further steps that the maximum is attained at only one index ℓ. The case that the maximum is attained simultaneously at two or more indeces ℓ (which almost never happens) requires more complications, which we would like to avoid here. One may refer to [25] for such details.

Now set $\Lambda_1 = \{\ell^{(1)}\}$. The vector $d^{(1)} \in \mathbb{R}^N$ describing the direction of the solution (homotopy) path has components

$$d^{(1)}_{\ell^{(1)}} = \|a_{\ell^{(1)}}\|_2^{-2} \operatorname{sign}((\Phi^* y)_{\ell^{(1)}}), \quad d^{(1)}_\ell = 0, \quad \ell \neq \ell^{(1)}.$$

The first linear piece of the solution path then takes the form

$$x = x(\gamma) = x^{(0)} + \gamma d^{(1)} = \gamma d^{(1)}, \quad \gamma \in [0, \gamma^{(1)}].$$

One verifies with the definition of $d^{(1)}$ that (28) is always satisfied for $x = x(\gamma)$ and $\lambda = \lambda(\gamma) = \lambda^{(0)} - \gamma$, $\gamma \in [0, \lambda^{(0)}]$. The next breakpoint is found by determining the maximal $\gamma = \gamma^{(1)} > 0$ for which (29) is satisfied, which is

$$\gamma^{(1)} = \min_{\ell \neq \ell^{(1)}} \left\{ \frac{\lambda^{(0)} - c_\ell^{(1)}}{1 - (\Phi^* \Phi d^{(1)})_\ell}, \frac{\lambda^{(0)} + c_\ell^{(1)}}{1 + (\Phi^* \Phi d^{(1)})_\ell} \right\}, \tag{34}$$

where the minimum is taken only over positive arguments, and there is a positive argument since the maximum in (33) is attained at only one index. Then $x^{(1)} = x(\gamma^{(1)}) = \gamma^{(1)} d^{(1)}$ is the next minimizer of $\mathcal{J}_\lambda$ for

$\lambda = \lambda^{(1)} := \lambda^{(0)} - \gamma^{(1)}$. This $\lambda^{(1)}$ satisfies $\lambda^{(1)} = \|c^{(1)}\|_{\ell_\infty}$. Let $\ell^{(2)}$ be the index where the minimum in (34) is attained (where we again assume that the minimum is attained only at one index) and put $\Lambda_2 = \{\ell^{(1)}, \ell^{(2)}\}$.

Step *j*: Determine the new direction $d^{(j)}$ of the homotopy path by solving

$$\Phi_{\Lambda_j}^* \Phi_{\Lambda_j} d_{\Lambda_j}^{(j)} = \operatorname{sign}(c_{\Lambda_j}^{(j)}), \tag{35}$$

which is a linear system of equations of size at most $|\Lambda_j| \times |\Lambda_j|$. Outside the components in Λ_j one sets $d_\ell^{(j)} = 0$, $\ell \notin \Lambda_j$. The next piece of the path is then given by

$$x(\gamma) = x^{(j-1)} + \gamma d^{(j)}, \quad \gamma \in [0, \gamma^{(j)}].$$

The maximal γ such that $x(\gamma)$ satisfies (29) is

$$\gamma_+^{(j)} = \min_{\ell \notin \Lambda_j} \left\{ \frac{\lambda^{(j-1)} - c_\ell^{(j)}}{1 - (\Phi^* \Phi d^{(j)})_\ell}, \frac{\lambda^{(j-1)} + c_\ell^{(j)}}{1 + (\Phi^* \Phi d^{(j)})_\ell} \right\}. \tag{36}$$

The maximal γ such that $x(\gamma)$ satisfies (28) is determined as

$$\gamma_-^{(j)} = \min_{\ell \in \Lambda_j} \{ -x_\ell^{(j-1)} / d_\ell^{(j)} \}. \tag{37}$$

Both in (36) and (37) the minimum is taken only over positive arguments. It may happen that all arguments in (37) are nonpositive. In this case, we set $\gamma_-^{(j)} = \infty$. The next breakpoint is given by $x^{(j+1)} = x(\gamma^{(j)})$ with $\gamma^{(j)} = \min\{\gamma_+^{(j)}, \gamma_-^{(j)}\}$. If $\gamma_+^{(j)}$ determines the minimum then the index $\ell_+^{(j)} \notin \Lambda_j$ providing the minimum in (36) is added to the active set, $\Lambda_{j+1} = \Lambda_j \cup \{\ell_+^{(j)}\}$. If $\gamma^{(j)} = \gamma_-^{(j)}$ then the index $\ell_-^{(j)} \in \Lambda_j$ is removed from the active set, $\Lambda_{j+1} = \Lambda_j \setminus \{\ell_-^{(j)}\}$. Further, one updates $\lambda^{(j)} = \lambda^{(j-1)} - \gamma^{(j)}$. By construction $\lambda^{(j)} = \|c^{(j)}\|_{\ell_\infty}$.

The algorithm stops when $\lambda^{(j)} = \|c^{(j)}\|_{\ell_\infty} = 0$, i.e., when the residual vanishes, and outputs $x^* = x^{(j)}$. Indeed, this happens after a finite number of steps. In [25] the authors proved the following result.

Theorem 5. *If in each step the minimum in* (36) *and* (37) *is attained in only one index ℓ, then the homotopy algorithm as described yields the minimizer of the ℓ_1-minimization problem* (4)*.*

If the algorithm is stopped earlier at some iteration j then obviously it yields the minimizer of $\mathcal{J}_\lambda = \mathcal{J}_{\lambda^{(j)}}$. In particular, obvious stopping rules may also be used to solve the problems BPDN (23) and LASSO (24).

The LARS (least angle regression) algorithm is a simple modification of the homotopy method, which only adds elements to the active set in each step. So $\gamma_-^{(j)}$ in (37) is not considered. (Sometimes the homotopy method is therefore also called modified LARS.) Clearly, LARS is not guaranteed any more to yield the solution of (4). However, it is observed empirically – and can be proven rigorously in certain cases [24] – that often in sparse recovery problems, the homotopy method does never remove elements from the active set, so that in this case LARS and homotopy perform the same steps. It is a crucial point that if the solution of (4) is k-sparse and the homotopy method never removes elements then the solution is obtained after precisely k-steps. Furthermore, the most demanding computational part at step j is then the solution of the $j \times j$ linear system of equations (35). In conclusion, the homotopy and LARS methods are very efficient for sparse recovery problems.

2.2 Iteratively Re-weighted Least Squares (IRLS)

Iteratively Re-weighted Least Squares (IRLS) is a method for solving minimization problems involving non-quadratic cost functions, perhaps non-convex and non-smooth, which however can be described as the infimum over a family of quadratic functions. This transformation suggests an algorithmic scheme that solves a sequence of quadratic problems, eventually tackled efficiently by tools of numerical linear algebra. Contrary to classical Newton methods the smoothness of the objective function is not needed in general. We refer to the recent paper [57] for an updated and rather general view about these methods.

In this section we want to present two different IRLS algorithms for sparse signal reconstruction. The first IRLS method addresses the standard ℓ_1-minimization (4). It was proposed in [16,34] and systematically analyzed in the works [13–15,22]. Besides a detailed description of the algorithm, the following subsection contains the proof that such algorithm has a guaranteed (local) linear rate of convergence which, with a minimal modification, can be improved to a superlinear rate.

In the second part of this section, we present an IRLS towards the solution of the ℓ_1-regularized least squares problem (26) and its convergence analysis. It was proposed independently in the works [46, 75], where also the convergence analysis has been carried out.

Despite its simplicity, versatility, and elegant analysis, IRLS does not outperform in general well-established first order methods, such as Iterative Hard Thresholding (IHT) [7] (see Section 2.3) or Fast Iterative Soft-Thresholding Algorithm (FISTA) [5] (see Section 3.1), as we also show in Section 4. In fact, its complexity very strongly depends on the way the solution of the successive quadratic optimizations is addressed, whether one uses preconditioned iterative methods and exploits fast matrix-vector multiplications or just considers simple direct linear solvers. If the dimensions of the problem are not too large or the involved matrices have no special structure allowing for fast matrix-vector multiplications, e.g., a partial circulant matrix or a partial Fourier matrix, then the use of a direct method such as Gaussian elimination can be appropriate. When instead the dimension of the problem is large and one can take advantage of the structure of the matrix to perform fast matrix-vector multiplications, then it would be convenient to use iterative solvers such as the Conjugate Gradient method (CG). The use of CG in the implementation of IRLS is appearing, for instance in [75] towards ℓ_1-norm minimization. However, the price to pay is that such solvers will return only an approximate solution whose precision depends on the number of iterations and a proper analysis of the convergence of the perturbed method. We extend the presentation of both methods in this section by such an analysis, which was recently elaborated in detail in [30], and clarify how accurately one needs to solve the quadratic problems by means of CG to guarantee both convergence and possibly also asymptotic super-linear rates.

2.2.1 IRLS method for ℓ_1-minimization

In this section we want to present an IRLS algorithm which, under the condition that Φ satisfies the NSP, is guaranteed to reconstruct vectors with the same approximation guarantees (5) as ℓ_1-minimization. Moreover, we will also show that such algorithm has a guaranteed (local) linear rate of convergence which, with a minimal modification, can be improved to a superlinear rate. We need to make first a brief introduction which hopefully will shed light on the basic principles of this algorithm and their interplay with sparse recovery and ℓ_1-minimization.

Denote again $\mathcal{F}_\Phi(y) = \{x : \Phi x = y\}$ and $\mathcal{N}_\Phi = \ker \Phi$. Let us start with a few non-rigorous observations; next we will be more precise. For $t \neq 0$ we simply have

$$|t| = \frac{t^2}{|t|}.$$

Hence, an ℓ_1-minimization can be recast into a weighted ℓ_2-minimization, and we may expect

$$\arg\min_{x\in\mathcal{F}_\Phi(y)} \sum_{j=1}^N |x_j| \approx \arg\min_{x\in\mathcal{F}_\Phi(y)} \sum_{j=1}^N x_j^2 |x_j^*|^{-1},$$

as soon as x^* is the wanted ℓ_1-norm minimizer (see [22, equation (1.4) and footnote 1] for a precise statement and proof). Clearly the advantage of this approximate reformulation is that minimizing a smooth quadratic function $|t|^2$ is better than addressing the minimization of the nonsmooth function $|t|$. However, the obvious drawbacks are that neither we dispose of x^* a priori (this is the vector we are interested to compute!) nor we can expect that $x_j^* \neq 0$ for all $j = 1,\dots,N$, since we hope for k-sparse solutions. Hence, we start assuming that we dispose of a good approximation w_j^n of $|(x_j^*)^2 + (\varepsilon^n)^2|^{-1/2} \approx |x_j^*|^{-1}$ and we compute

$$x^{n+1} = \arg\min_{x\in\mathcal{F}_\Phi(y)} \sum_{j=1}^N x_j^2 w_j^n, \tag{38}$$

then we up-date $\varepsilon^{n+1} \leq \varepsilon^n$, we define

$$w_j^{n+1} = |(x_j^{n+1})^2 + (\varepsilon^{n+1})^2|^{-1/2}, \tag{39}$$

and we iterate the process. The hope is that a proper choice of $\varepsilon^n \to 0$ will allow us for the computation of an ℓ_1-minimizer, although such a limit property is far from being obvious. The next sections will help us to describe the right mathematical setting where such limit is justified.

2.2.1.1 The IRLS algorithm

In the following we describe the essential lines of the analysis of the IRLS algorithm, as sketched in the introduction of Section 2.2.1 by taking advantage of the results and terminology already introduced in previous sections. Our analysis of the algorithm in (38) and (39) starts from the observation that

$$\frac{1}{|t|} = \min_{w>0} \frac{1}{2}\left(wt^2 + w^{-1}\right).$$

Inspired by this simple relationship, given a real number $\varepsilon > 0$ and a weight vector $w \in \mathbb{R}^N$, with $w_j > 0$, $j = 1,\dots,N$, we define

$$J(z,w,\varepsilon) := \frac{1}{2}\left[\sum_{j=1}^N z_j^2 w_j + \sum_{j=1}^N (\varepsilon^2 w_j + w_j^{-1})\right], \quad z \in \mathbb{R}^N. \tag{40}$$

The algorithm roughly described in (38) and (39) can be recast as an alternating minimization method for choosing competitors and weights based on the minimal properties of the functional J.

To describe this more rigorously, we recall that for $z \in \mathbb{R}^N$ the nonincreasing rearrangement $r(z)$ is the vector whose i-th entry $r(z)_i$ is the i-th largest element of the set $\{|z_j|,\ j = 1,\dots,N\}$, and a vector v is k-sparse if and only if $r(v)_{k+1} = 0$. We shall use the nonincreasing rearrangement to define a proper update rule for the approximation ε^n, see step 3 in Algorithm 1.

Algorithm 1 Iteratively Re-weighted Least Squares (IRLS)

Set $w^0 := (1,\dots,1)$, $\varepsilon^0 := 1$.

1: **while** $\varepsilon^n \neq 0$ **do**
2: $\quad x^{n+1} := \arg\min_{z \in \mathcal{F}_\Phi(y)} J(z, w^n, \varepsilon^n) = \arg\min_{z \in \mathcal{F}_\Phi(y)} \|z\|_{\ell_2(w^n)}$
3: $\quad \varepsilon^{n+1} := \min\left(\varepsilon^n, \frac{r(x^{n+1})_{K+1}}{N}\right)$
4: $\quad w^{n+1} := \arg\min_{w>0} J(x^{n+1}, w, \varepsilon^{n+1})$
5: **end while**

In general, the algorithm generates an infinite sequence $(x^n)_{n\in\mathbb{N}}$ of distinct vectors. However, if $\varepsilon^n = 0$ for some $n > 0$, define $x^\ell := x^n$ for $\ell > n$.

Each step of the algorithm requires the solution of a weighted least squares problem as in step 2 of Algorithm 1 (or simply (38)). In matrix form

$$x^{n+1} = D_n^{-1}\Phi^*(\Phi D_n^{-1}\Phi^*)^{-1}y, \tag{41}$$

where D_n is the $N \times N$ diagonal matrix whose j-th diagonal entry is w_j^n. Once x^{n+1} is computed and ε^{n+1} is updated as in step 3, the weight w^{n+1} is in fact given explicitly by (39).

2.2.1.2 Preliminary results

We now analyze some of the properties of the functional J defined by (40) along the iterations of the algorithm. Note that for each $n = 1,2,\dots$, we have

$$J(x^{n+1}, w^{n+1}, \varepsilon^{n+1}) = \sum_{j=1}^{N} [(x_j^{n+1})^2 + (\varepsilon^{n+1})^2]^{1/2}. \tag{42}$$

We also have the following monotonicity property which holds for all $n \geqslant 1$:

$$J(x^{n+1}, w^{n+1}, \varepsilon^{n+1}) \leqslant J(x^{n+1}, w^n, \varepsilon^{n+1}) \leqslant J(x^{n+1}, w^n, \varepsilon^n) \leqslant J(x^n, w^n, \varepsilon^n). \tag{43}$$

Here the first inequality follows from the minimization property that defines w^{n+1}, the second inequality from $\varepsilon^{n+1} \leqslant \varepsilon^n$, and the last inequality from the minimization property that defines x^{n+1}.

Lemma 5. *For each $n \geqslant 1$ we have*

$$\|x^n\|_{\ell_1} \leqslant J(x^1, w^0, \varepsilon^0) =: \mathcal{A} \tag{44}$$

and

$$w_j^n \geqslant \mathcal{A}^{-1}, \quad j = 1,\dots,N. \tag{45}$$

Proof. The bound (44) follows from (43) and

$$\|x^n\|_{\ell_1} \leqslant \sum_{j=1}^{N} [(x_j^n)^2 + (\varepsilon^n)^2]^{1/2} = J(x^n, w^n, \varepsilon^n).$$

The bound (45) follows from

$$(w_j^n)^{-1} = [(x_j^n)^2 + (\varepsilon^n)^2]^{1/2} \leqslant J(x^n, w^n, \varepsilon^n) \leqslant \mathcal{A},$$

where the last inequality uses (43). □

2.2.1.3 Convergence of the algorithm

Suppose that the weight w is *strictly positive* which we define to mean that $w_j > 0$ for all $j \in \{1, \ldots, N\}$. In this case, $\ell_2(w)$ is a Hilbert space with the inner product

$$\langle u, v \rangle_w := \sum_{j=1}^{N} w_j u_j v_j. \tag{46}$$

Define

$$x^w := \arg \min_{z \in \mathcal{F}_\Phi(y)} \|z\|_{\ell_2(w)}. \tag{47}$$

Because the $\|\cdot\|_{\ell_2(w)}$-norm is strictly convex, the minimizer x^w is necessarily unique; we leave as an easy exercise that x^w is completely characterized by the orthogonality conditions

$$\langle x^w, \eta \rangle_w = 0, \quad \text{for all } \eta \in \mathcal{N}_\Phi. \tag{48}$$

In this section, we prove that the algorithm converges. Our starting point is the following lemma that establishes $\|x^n - x^{n+1}\|_{\ell_2} \to 0$ for $n \to \infty$. Notice that this limit is already sufficient to state the "numerical convergence" as, after a sufficiently large amount of iterations, two consecutive iterations become indistinguishable.

Lemma 6. *Given any $y \in \mathbb{R}^m$, the x^n satisfy*

$$\sum_{n=1}^{\infty} \|x^{n+1} - x^n\|_{\ell_2}^2 \leqslant 2\mathcal{A}^2. \tag{49}$$

where $\mathcal{A}$ is the constant of Lemma 5. In particular, we have

$$\lim_{n \to \infty} \|x^n - x^{n+1}\|_{\ell_2} = 0. \tag{50}$$

Proof. For each $n = 1, 2, \ldots$, we have

$$2[J(x^n, w^n, \varepsilon^n) - J(x^{n+1}, w^{n+1}, \varepsilon^{n+1})] \geqslant 2[J(x^n, w^n, \varepsilon^n) - J(x^{n+1}, w^n, \varepsilon^n)] = \langle x^n, x^n \rangle_{w^n} - \langle x^{n+1}, x^{n+1} \rangle_{w^n}$$
$$= \langle x^n + x^{n+1}, x^n - x^{n+1} \rangle_{w^n} = \langle x^n - x^{n+1}, x^n - x^{n+1} \rangle_{w^n} = \sum_{j=1}^{N} w_j^n (x_j^n - x_j^{n+1})^2 \geqslant \mathcal{A}^{-1} \|x^n - x^{n+1}\|_{\ell_2}^2, \tag{51}$$

where the third equality uses the fact that $\langle x^{n+1}, x^n - x^{n+1}\rangle_{w^n} = 0$ (observe that $x^{n+1} - x^n \in \mathcal{N}_\Phi$ and invoke (48)), and the inequality uses the bound (45) on the weights. If we now sum these inequalities over $n \geqslant 1$, we arrive at (49). □

A reader not interested to a precise statement of convergence may want at this point to directly move to Section 2.2.1.4 where the rate of local convergence is analyzed.

From the monotonicity of ε^n, we know that $\varepsilon := \lim_{n\to\infty} \varepsilon^n$ exists and is non-negative. The following functional will play an important role in our proof of convergence:

$$f_\varepsilon(z) := \sum_{j=1}^{N} (z_j^2 + \varepsilon^2)^{1/2}. \tag{52}$$

Notice that if we knew that x^n converged to x then, in view of (42), $f_\varepsilon(x)$ would be the limit of $J(x^n, w^n, \varepsilon^n)$. When $\varepsilon > 0$ the functional f_ε is strictly convex and therefore has a unique minimizer

$$x^\varepsilon := \arg\min_{z\in\mathcal{F}_\Phi(y)} f_\varepsilon(z). \tag{53}$$

This minimizer is characterized by the following lemma:

Lemma 7. *Let $\varepsilon > 0$ and $z \in \mathcal{F}_\Phi(y)$. Then $z = x^\varepsilon$ if and only if $\langle z, \eta\rangle_{\widetilde{w}(z,\varepsilon)} = 0$ for all $\eta \in \mathcal{N}_\Phi$, where $\widetilde{w}(z,\varepsilon)_j = [z_j^2 + \varepsilon^2]^{-1/2}$.*

Proof. A proof of this Lemma is given in [22, Lemma 5.2]. □

We now prove the convergence of the algorithm.

Theorem 6. *Let K (the same index as used in the ε-update rule in step 3 of Algorithm IRLS) be chosen so that Φ satisfies the Null Space Property of order K, with $\gamma < 1$. Then, for each $y \in \mathbb{R}^m$, the output of Algorithm IRLS converges to a vector $\bar{x}$, with $r(\bar{x})_{K+1} = N\lim_{n\to\infty}\varepsilon^n$ and the following hold:*
(i) *If $\varepsilon = \lim_{n\to\infty}\varepsilon^n = 0$, then $\bar{x}$ is K-sparse; in this case there is therefore a unique ℓ_1-minimizer x^*, and $\bar{x} = x^*$; moreover, we have, for $k \leqslant K$, and any $z \in \mathcal{F}_\Phi(y)$,*

$$\|z - \bar{x}\|_{\ell_1} \leqslant c\sigma_k(z)_{\ell_1}, \quad \text{with } c := \frac{2(1+\gamma)}{1-\gamma} \tag{54}$$

(ii) *If $\varepsilon = \lim_{n\to\infty}\varepsilon^n > 0$, then $\bar{x} = x^\varepsilon$;*
(iii) *In this last case, if γ satisfies the stricter bound $\gamma < 1 - \frac{2}{K+2}$ (or, equivalently, if $\frac{2\gamma}{1-\gamma} < K$), then we have, for all $z \in \mathcal{F}_\Phi(y)$ and any $k < K - \frac{2\gamma}{1-\gamma}$, that*

$$\|z - \bar{x}\|_{\ell_1} \leqslant \tilde{c}\sigma_k(z)_{\ell_1}, \quad \text{with } \tilde{c} := \frac{2(1+\gamma)}{1-\gamma}\left[\frac{K-k+\frac{3}{2}}{K-k-\frac{2\gamma}{1-\gamma}}\right] \tag{55}$$

As a consequence, this case is excluded if $\mathcal{F}_\Phi(y)$ contains a vector of sparsity $k < K - \frac{2\gamma}{1-\gamma}$.

Note that the approximation properties (54) and (55) are exactly of the same order as the one (5) provided by ℓ_1-minimization. However, in general, $\bar{x}$ is not necessarily an ℓ_1-minimizer, unless it coincides with a sparse

solution.
The constant $\tilde{c}$ can be quite reasonable; for instance, if $\gamma \leqslant 1/2$ and $k \leqslant K-3$, then we have $\tilde{c} \leqslant 9\,\frac{1+\gamma}{1-\gamma} \leqslant 27$.

Proof. Note that since $\varepsilon^{n+1} \leq \varepsilon^n$, the ε^n always converge. We start by considering the case $\varepsilon := \lim_{n\to\infty} \varepsilon^n = 0$.

Case $\varepsilon = 0$: In this case, we want to prove that x^n converges , and that its limit is an ℓ_1-minimizer. Suppose that $\varepsilon^{n_0} = 0$ for some n_0. Then by the definition of the algorithm, we know that the iteration is stopped at $n = n_0$, and $x^n = x^{n_0}$, $n \geqslant n_0$. Therefore $\bar{x} = x^{n_0}$. From the definition of ε^n, it then also follows that $r(x^{n_0})_{K+1} = 0$ and so $\bar{x} = x^{n_0}$ is K-sparse. As noted in [22, Lemma 4.3], if a K-sparse solution exists when Φ satisfies the NSP of order K with $\gamma < 1$, then it is the unique ℓ_1-minimizer. Therefore, $\bar{x}$ equals x^*, this unique minimizer.

Suppose now that $\varepsilon^n > 0$ for all n. Since $\varepsilon^n \to 0$, there is an increasing sequence of indices (n_i) such that $\varepsilon^{n_i} < \varepsilon^{n_i-1}$ for all i. By the definition (3) of $(\varepsilon^n)_{n\in\mathbb{N}}$, we must have $r(x^{n_i})_{K+1} < N\varepsilon^{n_i-1}$ for all i. Noting that $(x^n)_{n\in\mathbb{N}}$ is a bounded sequence, there exists a subsequence $(p_j)_{j\in\mathbb{N}}$ of $(n_i)_{i\in\mathbb{N}}$ such that $(x^{p_j})_{j\in\mathbb{N}}$ converges to a point $\widetilde{x} \in \mathcal{F}_\Phi(y)$. By [22, Lemma 4.1], we know that $r(x^{p_j})_{K+1}$ converges to $r(\widetilde{x})_{K+1}$. Hence we get

$$r(\widetilde{x})_{K+1} = \lim_{j\to\infty} r(x^{p_j})_{K+1} \leqslant \lim_{j\to\infty} N\varepsilon^{p_j-1} = 0, \tag{56}$$

which means that the support-width of $\widetilde{x}$ is at most K, i.e. $\widetilde{x}$ is K-sparse. By the same token used above, we again have that $\widetilde{x} = x^*$, the unique ℓ_1-minimizer. We must still show that $x^n \to x^*$. Since $x^{p_j} \to x^*$ and $\varepsilon^{p_j} \to 0$, (42) implies $J(x^{p_j}, w^{p_j}, \varepsilon^{p_j}) \to \|x^*\|_{\ell_1}$. By the monotonicity property stated in (43), we get $J(x^n, w^n, \varepsilon^n) \to \|x^*\|_{\ell_1}$. Since (42) implies

$$J(x^n, w^n, \varepsilon^n) - N\varepsilon^n \leqslant \|x^n\|_{\ell_1} \leqslant J(x^n, w^n, \varepsilon^n), \tag{57}$$

we obtain $\|x^n\|_{\ell_1} \to \|x^*\|_{\ell_1}$. Finally, we invoke [22, Lemma 4.2] with $z' = x^n$, $z = x^*$, and $k = K$ to get

$$\limsup_{n\to\infty} \|x^n - x^*\|_{\ell_1} \leqslant \frac{1+\gamma}{1-\gamma}\left(\lim_{n\to\infty} \|x^n\|_{\ell_1} - \|x^*\|_{\ell_1}\right) = 0, \tag{58}$$

which completes the proof that $x^n \to x^*$ in this case.

Finally, (54) follows from [22, Lemma 4.3] (with $L = K$), and the observation that $\sigma_n(z) \geqslant \sigma_{n'}(z)$ if $n \leqslant n'$.

Case $\varepsilon > 0$: We shall first show that $x^n \to x^\varepsilon$, $n \to \infty$, with x^ε as defined by (53). By Lemma 5, we know that $(x^n)_{n=1}^\infty$ is a bounded sequence in $\mathbb{R}^N$ and hence this sequence has accumulation points. Let (x^{n_i}) be any convergent subsequence of (x^n) and let $\widetilde{x} \in \mathcal{F}_\Phi(y)$ be its limit. We want to show that $\widetilde{x} = x^\varepsilon$.

Since $w_j^n = [(x_j^n)^2 + (\varepsilon^n)^2]^{-1/2} \leqslant \varepsilon^{-1}$, it follows that $\lim_{i\to\infty} w_j^{n_i} = [(\widetilde{x}_j)^2 + \varepsilon^2]^{-1/2} = \widetilde{w}(\widetilde{x}, \varepsilon)_j =: \widetilde{w}_j$, $j = 1, \ldots, N$. On the other hand, by invoking Lemma 6, we now find that $x^{n_i+1} \to \widetilde{x}$, $i \to \infty$. It then follows from the orthogonality relations (48) that for every $\eta \in \mathcal{N}_\Phi$, we have

$$\langle \widetilde{x}, \eta \rangle_{\widetilde{w}} = \lim_{i\to\infty} \langle x^{n_i+1}, \eta \rangle_{w^{n_i}} = 0. \tag{59}$$

Now the "if" part of Lemma 7 implies that $\widetilde{x} = x^\varepsilon$. Hence x^ε is the unique accumulation point of $(x^n)_{n\in\mathbb{N}}$ and therefore its limit. This establishes (ii).

To prove the error estimate (55) stated in (iii), we first note that for any $z \in \mathcal{F}_\Phi(y)$, we have

$$\|x^\varepsilon\|_{\ell_1} \leqslant f_\varepsilon(x^\varepsilon) \leqslant f_\varepsilon(z) \leqslant \|z\|_{\ell_1} + N\varepsilon, \tag{60}$$

where the second inequality uses the minimizing property of x^ε. Hence it follows that $\|x^\varepsilon\|_{\ell_1} - \|z\|_{\ell_1} \leqslant N\varepsilon$. We now invoke [22, Lemma 4.2] to obtain

$$\|x^\varepsilon - z\|_{\ell_1} \leqslant \frac{1+\gamma}{1-\gamma}[N\varepsilon + 2\sigma_k(z)_{\ell_1}]. \tag{61}$$

From [22, Lemma 4.1] and (3), we obtain

$$N\varepsilon = \lim_{n\to\infty} N\varepsilon^n \leqslant \lim_{n\to\infty} r(x^n)_{K+1} = r(x^\varepsilon)_{K+1}. \tag{62}$$

It furthermore follows from [22, Lemma 4.1] that

$$\begin{aligned}(K+1-k)N\varepsilon &\leqslant (K+1-k)r(x^\varepsilon)_{K+1}\\ &\leqslant \|x^\varepsilon - z\|_{\ell_1} + \sigma_k(z)_{\ell_1}\\ &\leqslant \frac{1+\gamma}{1-\gamma}[N\varepsilon + 2\sigma_k(z)_{\ell_1}] + \sigma_k(z)_{\ell_1},\end{aligned} \tag{63}$$

where the last inequality uses (61). Since by assumption on K, we have $K-k > \frac{2\gamma}{1-\gamma}$, i.e. $K+1-k > \frac{1+\gamma}{1-\gamma}$, we obtain

$$N\varepsilon + 2\sigma_k(z)_{\ell_1} \leqslant \frac{2(K-k)+3}{(K-k)-\frac{2\gamma}{1-\gamma}}\,\sigma_k(z)_{\ell_1}.$$

Using this back in (61), we arrive at (55).

Finally, notice that if $\mathcal{F}_\Phi(y)$ contains a k-sparse vector (with $k < K - \frac{2\gamma}{1-\gamma}$), then we know already that this must be the unique ℓ_1-minimizer x^*; it then follows from our arguments above that we must have $\varepsilon = 0$. Indeed, if we had $\varepsilon > 0$, then (63) would hold for $z = x^*$; since x^* is k-sparse, $\sigma_k(x^*)_{\ell_1} = 0$, implying $\varepsilon = 0$, a contradiction with the assumption $\varepsilon > 0$. This finishes the proof. □

2.2.1.4 Local linear rate of convergence

It is instructive to show a further very interesting result concerning the local rate of convergence of this algorithm, which makes heavy use of the NSP as well as the optimality properties we introduced above. One assumes here that $\mathcal{F}_\Phi(y)$ contains a k-sparse vector x^*. The algorithm produces the sequence x^n, which converges to x^*, as established above. One denotes the (unknown) support of the k-sparse vector x^* by Λ.

We introduce an auxiliary sequence of error vectors $\eta^n \in \mathcal{N}_\Phi$ via $\eta^n := x^n - x^*$ and

$$E_n := \|\eta^n\|_{\ell_1} = \|x^* - x^n\|_{\ell_1}.$$

We know that $E_n \to 0$.

The following theorem gives a bound on the rate of convergence of E_n to zero.

Theorem 7. *Assume A satisfies NSP of order K with constant γ such that $0 < \gamma < 1 - \frac{2}{K+2}$. Suppose that $k < K - \frac{2\gamma}{1-\gamma}$, $0 < \rho < 1$, and $0 < \gamma < 1 - \frac{2}{K+2}$ are such that*

$$\mu := \frac{\gamma(1+\gamma)}{1-\rho}\left(1 + \frac{1}{K+1-k}\right) < 1.$$

Assume that $\mathcal{F}_\Phi(y)$ contains a k-sparse vector x^* and let $\Lambda = \operatorname{supp}(x^*)$. Let n_0 be such that

$$E_{n_0} \leqslant R^* := \rho \min_{j\in\Lambda} |x_j^*|. \tag{64}$$

Then for all $n \geqslant n_0$, we have

$$E_{n+1} \leqslant \mu\, E_n. \tag{65}$$

Consequently x^n converges to x^* exponentially.

Proof. We start with the relation (48) with $w = w^n$, $x^w = x^{n+1} = x^* + \eta^{n+1}$, and $\eta = x^{n+1} - x^* = \eta^{n+1}$, which gives

$$\sum_{j=1}^{N} (x_j^* + \eta_j^{n+1})\eta_j^{n+1} w_j^n = 0.$$

Rearranging the terms and using the fact that x^* is supported on Λ, we get

$$\sum_{j=1}^{N} |\eta_j^{n+1}|^2 w_j^n = -\sum_{j\in\Lambda} x_j^* \eta_j^{n+1} w_j^n = -\sum_{j\in\Lambda} \frac{x_j^*}{[(x_j^n)^2 + (\varepsilon^n)^2]^{1/2}} \eta_j^{n+1}. \tag{66}$$

Prove of the theorem is by induction. One assumes that we have shown $E_n \leqslant R^*$ already. We then have, for all $j \in \Lambda$,

$$|\eta_j^n| \leqslant \|\eta^n\|_{\ell_1} = E_n \leqslant \rho |x_j^*| \ ,$$

so that

$$\frac{|x_j^*|}{[(x_j^n)^2 + (\varepsilon^n)^2]^{1/2}} \leqslant \frac{|x_j^*|}{|x_j^n|} = \frac{|x_j^*|}{|x_j^* + \eta_j^n|} \leqslant \frac{1}{1-\rho}, \tag{67}$$

and hence (66) combined with (67) and NSP gives

$$\sum_{j=1}^{N} |\eta_j^{n+1}|^2 w_j^n \leqslant \frac{1}{1-\rho} \|\eta_\Lambda^{n+1}\|_{\ell_1} \leqslant \frac{\gamma}{1-\rho} \|\eta_{\Lambda^c}^{n+1}\|_{\ell_1}.$$

At the same time, the Cauchy-Schwarz inequality combined with the above estimate yields

$$\begin{aligned} \|\eta_{\Lambda^c}^{n+1}\|_{\ell_1}^2 &\leqslant \left(\sum_{j\in\Lambda^c} |\eta_j^{n+1}|^2 w_j^n\right) \left(\sum_{j\in\Lambda^c} [(x_j^n)^2 + (\varepsilon^n)^2]^{1/2}\right) \leqslant \left(\sum_{j=1}^{N} |\eta_j^{n+1}|^2 w_j^n\right) \left(\sum_{j\in\Lambda^c} [(\eta_j^n)^2 + (\varepsilon^n)^2]^{1/2}\right) \\ &\leqslant \frac{\gamma}{1-\rho} \|\eta_{\Lambda^c}^{n+1}\|_{\ell_1} \left(\|\eta^n\|_{\ell_1} + N\varepsilon^n\right). \end{aligned} \tag{68}$$

If $\eta_{\Lambda^c}^{n+1} = 0$, then $x_{\Lambda^c}^{n+1} = 0$. In this case x^{n+1} is k-sparse and the algorithm has stopped by definition; since $x^{n+1} - x^*$ is in the null space $\mathcal{N}_\Phi$, which contains no k-sparse elements other than 0, we have already obtained the solution $x^{n+1} = x^*$. If $\eta_{\Lambda^c}^{n+1} \neq 0$, then after canceling the factor $\|\eta_{\Lambda^c}^{n+1}\|_{\ell_1}$ in (68), we get

$$\|\eta_{\Lambda^c}^{n+1}\|_{\ell_1} \leqslant \frac{\gamma}{1-\rho} \left(\|\eta^n\|_{\ell_1} + N\varepsilon^n\right),$$

and thus

$$\|\eta^{n+1}\|_{\ell_1} = \|\eta_\Lambda^{n+1}\|_{\ell_1} + \|\eta_{\Lambda^c}^{n+1}\|_{\ell_1} \leqslant (1+\gamma)\|\eta_{\Lambda^c}^{n+1}\|_{\ell_1} \leqslant \frac{\gamma(1+\gamma)}{1-\rho} \left(\|\eta^n\|_{\ell_1} + N\varepsilon^n\right). \tag{69}$$

Now, we also have by (3) and and application of [22, Lemma 4.1]

$$N\varepsilon^n \leqslant r(x^n)_{K+1} \leqslant \frac{1}{K+1-k}(\|x^n - x^*\|_{\ell_1} + \sigma_k(x^*)_{\ell_1}) = \frac{\|\eta^n\|_{\ell_1}}{K+1-k}, \tag{70}$$

since by assumption $\sigma_k(x^*) = 0$. This, together with (69), yields the desired bound,

$$E_{n+1} = \|\eta^{n+1}\|_{\ell_1} \leqslant \frac{\gamma(1+\gamma)}{1-\rho}\left(1 + \frac{1}{K+1-k}\right)\|\eta^n\|_{\ell_1} = \mu E_n.$$

In particular, since $\mu < 1$, we have $E_{n+1} \leqslant R^*$, which completes the induction step. It follows that $E_{n+1} \leqslant \mu E_n$ for all $n \geqslant n_0$. □

2.2.1.5 Conjugate gradient acceleration of the IRLS method for ℓ_1-minimization

Recall that we mentioned earlier in this section that, besides its simplicity and intuitive approach, an implementation of Algorithm IRLS is in general not competitive to first order methods as Iterative Hard Thresholding (IHT) or Fast Iterative Soft Thresholding Algorithm (FISTA). This is doubtlessly confirmed by the numerical experiments which are presented in Section 4. In this section, we present a modified version of IRLS by insertion of a Conjugate Gradient method for the solution of the successive quadratic optimizations, and two results providing separately the convergence and the rate of convergence of the modified algorithm. In particular, we explain how to control the inaccuracies which appear as a consequence of inexact solutions of the intermediate least squares problems. The proofs of the theorems are sketched. This modification allows the method to perform at least as good as the aforementioned competitors, to be faster in some cases, and usually way more robust in general.

Instead of solving *exactly* the system of linear equations occurring in step 2 of Algorithm IRLS as in formula (41), we substitute the exact solution by an approximate solution provided by the iterative Modified Conjugate Gradient Algorithm (MCG) as described in Appendix A. We shall set a tolerance tol_{n+1}, which gives us an upper threshold for the error between the optimal and the approximate solution in the weighted ℓ_2-norm. In this section, we give a precise and implementable condition on the sequence $(\mathrm{tol}_n)_{n\in\mathbb{N}}$ of the tolerances to guarantee the convergence of the algorithm.

Algorithm 2 Iteratively Re-weighted Least Squares combined with CG (CG-IRLS)

Set $w^0 := (1,\dots,1)$, $\varepsilon^0 := 1$, $\beta \in (0,1]$

1: **while** $\varepsilon^n \neq 0$ **do**
2: Compute $\tilde{x}^{n+1}$ by means of MCG s.t. $\|\tilde{x}^{n+1} - \hat{x}^{n+1}\|^2_{\ell_2(w^n)} \leq \mathrm{tol}_{n+1}$,
where $\hat{x}^{n+1} := \operatorname{arg\,min}_{x\in\mathcal{F}_\Phi(y)} J(x, w^n, \varepsilon^n) = \operatorname{arg\,min}_{z\in\mathcal{F}_\Phi(y)} \|z\|_{\ell_2(w^n)}$
3: $\varepsilon^{n+1} := \min(\varepsilon^n, \beta r(\tilde{x}^{n+1})_{K+1})$
4: $w^{n+1} := \operatorname{arg\,min}_{w>0} J(\tilde{x}^{n+1}, w, \varepsilon^{n+1})$, i.e., $w_j^{n+1} = [|\tilde{x}_j^{n+1}|^2 + (\varepsilon^{n+1})^2]^{-\frac{1}{2}}, \qquad j = 1,\dots,N$
5: **end while**

In contrast to Algorithm IRLS, the value β in step 3 is introduced to obtain flexibility in tuning the performance of the algorithm. While we prove in Theorem 8 convergence for any positive value of β, we only have the instance optimality in Theorem 8(iii) for $\beta < \frac{1-\gamma}{1+\gamma}\frac{K+1-k}{N}$ in the case that the $\lim_{n\to\infty} \varepsilon^n \neq 0$. However,

in the numerical experiments of Section 4, we investigate best choices of β which are actually not fulfilling this restriction.

From now on, we fix the notation $\hat{x}^{n+1}$ for the exact solution in step 2 of Algorithm CG-IRLS, and $\tilde{x}^{n+1,i}$ for its approximate solution in the i-th iteration of Algorithm MCG (see Appendix A). We have to make sure that $\|\hat{x}^{n+1}-\tilde{x}^{n+1,i}\|^2_{\ell_2(w^n)}$ is sufficiently small to fall below the given tolerance. To this end, we could use the bound on the error provided by (157), but it has the following two unpractical drawbacks:

1. The vector $\hat{x}=\hat{x}^{n+1}$ is not not known a priori;
2. The computation of the condition number c_{TT^*} is possible, but it requires the computation of the eigenvalues with additional substantial computational cost which we prefer to avoid.

Hence, we propose an alternative estimate of the error. We use the notation of Algorithm MCG, but we add an additional upper index for the outer IRLS iteration, e.g., $\theta^{n+1,i}$ is the θ^i in the $n+1$-th IRLS iteration. After i steps of MCG we have

$$\|\hat{x}^{n+1}-\tilde{x}^{n+1,i}\|^2_{\ell_2(w^n)} = \|D_n\Phi^*(\Phi D_n\Phi^*)^{-1}y - D_n\Phi^*\theta^{n+1,i}\|^2_{\ell_2(w^n)}.$$

We use $\theta^{n+1,i}=(\Phi D_n\Phi^*)^{-1}(y-\rho^{n+1,i})$ from step 5 of MCG to obtain

$$\begin{aligned}\|\hat{x}^{n+1}-\tilde{x}^{n+1,i}\|^2_{\ell_2(w^n)} &= \|D_n^{\frac{1}{2}}\Phi^*(\Phi D_n\Phi^*)^{-1}\rho^{n+1,i}\|^2_{\ell_2} \le \|D_n\|\|\Phi\|^2\|(\Phi D_n\Phi^*)^{-1}\|^2\|\rho^{n+1,i}\|^2_{\ell_2}\\ &= \frac{\max\limits_{i\in\mathbb{N}}\left(|\tilde{x}_i|^2+(\varepsilon^n)^2\right)^{\frac{1}{2}}\|\Phi\|^2}{\lambda_{\min}(\Phi D_n\Phi^*)}\|\rho^{n+1,i}\|^2_{\ell_2} \le \left(1+\max_{1\le i\le N}\left(\frac{|\tilde{x}_i^n|}{\varepsilon^n}\right)^2\right)^{\frac{1}{2}}\frac{\|\Phi\|^2}{\sigma_{\min}(\Phi)}\|\rho^{n+1,i}\|^2_{\ell_2}.\end{aligned}$$

The last inequality above is the result from $\lambda_{\min}(\Phi D_n\Phi^*)=\sigma^2_{\min}\left(\Phi D_n^{\frac{1}{2}}\right)$ and

$$\sigma_{\min}\left(\Phi D_n^{\frac{1}{2}}\right)=\sigma_{\min}(\Phi)\,\sigma_{\min}\left(D_n^{\frac{1}{2}}\right)\ge(\varepsilon^n)^{2-\tau}\sigma_{\min}(\Phi),$$

where $\lambda_{\min}(\cdot)$ and $\sigma_{\min}(\cdot)$ denote the smallest eigenvalue and singular value respectively. Since ε^n and $\tilde{x}^n$ are known from the previous iteration, and $\|\rho^{n+1,i}\|_{\ell_2}$ is explicitly calculated within the MCG algorithm, $\|\hat{x}^{n+1}-\tilde{x}^{n+1,i}\|^2_{\ell_2(w^n)}\le \mathrm{tol}_{n+1}$ can be achieved by iterating until

$$\|\rho^{n+1,i}\|^2_{\ell_2}\le\frac{\sigma_{\min}(\Phi)}{\left(1+\max\limits_{1\le i\le N}\left(\frac{|\tilde{x}_i^n|}{\varepsilon^n}\right)^2\right)^{\frac{1}{2}}\|\Phi\|^2}\mathrm{tol}_{n+1}. \tag{71}$$

Consequently, we shall iterate until the minimal iteration number $i\in\mathbb{N}$ is such that the above inequality is fulfilled and set $\tilde{x}^{n+1}:=\tilde{x}^{n+1,i}$, which will be the standard notation for the approximate solution.

In inequality (71), the computation of $\sigma_{\min}(\Phi)$ and $\|\Phi\|$ is necessary. The computation of these constants might be demanding, but has to be performed only once before the algorithm starts. Furthermore, in practice it is sufficient to compute approximations of these values. Due to these reasons, these operations are not critical for the computation time of the algorithm.

After introducing Algorithm CG-IRLS, we state below the two main results of this section. Theorem 8 shows the convergence of the algorithm to a limit point, with certain error guarantees with respect to the solution of (4).

Theorem 8. *Let K (the same index as used in the ε-update rule in step 3 of Algorithm CG-IRLS) be chosen such that Φ satisfies the Null Space Property* (9) *of order K, with $\gamma < 1$. If tol_{n+1} in Algorithm CG-IRLS is chosen such that*

$$\sqrt{\mathrm{tol}_{n+1}} \leq \sqrt{\left(\frac{c_n}{2}\right)^2 + \frac{2a_{n+1}}{\tau \bar{W}_{n+1}^2}} - \frac{c_n}{2}, \tag{72}$$

where

$$c_n := 2W_n \left(\|\tilde{x}^n\|_{\ell_2(w^{n-1})} + \sqrt{\mathrm{tol}_n} \right), \text{ with} \tag{73}$$

$$\bar{W}_n := \sqrt{\frac{\max_i |\tilde{x}_i^{n-1}| + (\varepsilon^{n-1})}{(\varepsilon^n)}}, \text{ and } W_n := \left\| D_n^{-\frac{1}{2}} D_{n-1}^{\frac{1}{2}} \right\|,$$

for a sequence $(a_n)_{n\in\mathbb{N}}$, which fulfills $a_n \geq 0$ for all $n \in \mathbb{N}$, and $\sum_{i=0}^{\infty} a_n < \infty$, then, for each $y \in \mathbb{R}^m$, Algorithm CG-IRLS produces a non-empty set of accumulation points $\mathcal{Z}(y)$. Define $\varepsilon := \lim_{n\to\infty} \varepsilon^n$, then the following holds:

(i) If $\varepsilon = 0$, then $\mathcal{Z}(y)$ consists of a single K-sparse vector $\bar{x}$, which is the unique ℓ_1-minimizer in $\mathcal{F}_\Phi(y)$. Moreover, we have, for $k \leq K$ and any $x \in \mathcal{F}_\Phi(y)$:

$$\|x - \bar{x}\|_{\ell_1} \leq c_1 \sigma_k(x)_{\ell_1}, \quad \text{with } c_1 := 2\frac{1+\gamma}{1-\gamma}. \tag{74}$$

(ii) If $\varepsilon > 0$, then $\mathcal{Z}(y)$ consists of a single element $\bar{x}$, and $\bar{x} = x^\varepsilon = \arg\min_{z\in\mathcal{F}_\Phi(y)} f_\varepsilon(z)$ (compare (53)*), and we have $\langle \bar{x}, \eta \rangle_{\bar{w}} = 0$, with $\bar{w} := \left[\left(\bar{x}_i^2 + \varepsilon^2\right)^{-\frac{1}{2}} \right]_{i=1}^N$, for all $\eta \in \mathcal{N}_\Phi$. Moreover, for each $x \in \mathcal{F}_\Phi(y)$ and any $\beta < \frac{1-\gamma}{1+\gamma}\frac{K+1-k}{N}$, we have*

$$\|x - \bar{x}\|_{\ell_1} \leq c_2 \sigma_k(x)_{\ell_1}, \quad \text{with } c_2 := \frac{1+\gamma}{1-\gamma}\left(\frac{2 + \frac{N\beta}{K+1-k}}{1 - \frac{N\beta}{K+1-k}\frac{1+\gamma}{1-\gamma}} \right). \tag{75}$$

Knowing that the algorithm converges and leads to an adequate solution eventually is a good start, but one is also naturally interested in how fast one gets to this solution. Theorem 9 states that a linear rate of convergence can be established in a ball around the solution.

Theorem 9. *Assume Φ satisfies the NSP of order K with constant γ such that $0 < \gamma < 1 - \frac{2}{K+2}$, and that $\mathcal{F}_\Phi(y)$ contains a k-sparse vector x^*. Define $\Lambda := \mathrm{supp}(x^*)$. Suppose that $k < K - \frac{2\gamma}{1-\gamma}$ and $0 < \rho < 1$ are such that*

$$\begin{aligned}\mu &:= \frac{\gamma(1+\gamma)}{(1-\rho)}\left(1+\frac{(N-k)\beta}{K+1-k}\right) < 1,\\ R^* &:= \rho \min_{j\in\Lambda} |x_j^*|,\end{aligned} \tag{76}$$

and chose $\tilde{\mu}$ *such that* $\mu < \tilde{\mu} < 1$. *Define the error*

$$E_n := \|\tilde{x}^n - x^*\|_{\ell_1}. \tag{77}$$

Assume there exists n_0 *such that*

$$E_{n_0} \leq R^*. \tag{78}$$

If a_{n+1} *and* tol_{n+1} *are chosen as in Theorem 8 with the additional bound*

$$\mathrm{tol}_{n+1} \leq \frac{((\tilde{\mu}-\mu)E_n)^2}{NC}, \tag{79}$$

then for all $n \geq n_0$, *we have*

$$E_{n+1} \leq \mu E_n + \sqrt{NC\,\mathrm{tol}_{n+1}}, \tag{80}$$

and

$$E_{n+1} \leq \tilde{\mu} E_n, \tag{81}$$

where $C := 3\sum_{n=1}^{\infty} a_n + J\left(\tilde{x}^1, w^0, \varepsilon^0\right)$. *Consequently* $\tilde{x}^n$ *converges locally linearly to* x^*.

Remark 1. Note that the second bound in (79), which implies (81), is only of theoretical nature. Since the value of E_n is unknown it cannot be computed in an implementation. However, heuristic choices of tol_{n+1} may fulfill this bound. Thus, in practice one can only guarantee the "asymptotic" linear convergence (80).

In the remainder of this section we aim to sketch the proofs of both results. The interested reader is referred to [30] for a more detailed presentation.

Proof (Theorem 8). The proof follows roughly the same steps as the proof of Theorem 6. We will sketch the main steps here and highlight in particular the changes that have to be made due to step 2 in Algorithm CG-IRLS, where only an approximation of the weighted least squares minimizer is computed.

The most important difference towards the analysis of the convergence of Algorithm IRLS is the fact that from the monotonicity property (42) only

$$J\left(\tilde{x}^{n+1}, w^{n+1}, \varepsilon^{n+1}\right) \leq J\left(\tilde{x}^{n+1}, w^n, \varepsilon^{n+1}\right) \leq J\left(\tilde{x}^{n+1}, w^n, \varepsilon^n\right) \tag{82}$$

holds. However, in general

$$J\left(\tilde{x}^{n+1}, w^n, \varepsilon^n\right) \not\leq J\left(\tilde{x}^n, w^n, \varepsilon^n\right).$$

Instead, we are only able to show that for a tolerance tol_n, which is chosen as in (72) depending on a given positive scalar a_n, that

$$\left|J\left(\hat{x}^{n+1},w^{n+1},\varepsilon^{n+1}\right)-J\left(\tilde{x}^{n+1},w^{n+1},\varepsilon^{n+1}\right)\right|\leq a_{n+1}, \tag{83}$$

$$\left|J\left(\hat{x}^{n+1},w^{n},\varepsilon^{n}\right)-J\left(\tilde{x}^{n+1},w^{n},\varepsilon^{n}\right)\right|\leq a_{n+1}, \text{ and} \tag{84}$$

$$J\left(\hat{x}^{n+1},w^{n+1},\varepsilon^{n+1}\right)\leq J\left(\hat{x}^{n+1},w^{n},\varepsilon^{n}\right)+2a_{n+1}, \tag{85}$$

where we want to remind that $\hat{x}^{n+1}=\underset{x\in\mathcal{F}_\Phi(y)}{\arg\min}\, J\left(x,w^n,\varepsilon^n\right)$ (see [30, Lemma 4]). By means of these collection of estimates and the summability of the sequence $(a_n)_{n\in\mathbb{N}}$ one can show that there is an upper bound of the sequence $\tilde{x}^n$ and a lower bound of the sequence w_j^n, $j=1,\dots,N$, similarly to Lemma 5. Furthermore one can also show that the difference between the successive iterates as well as the exact and approximated solution becomes arbitrarily small (see [30, Lemma 6, Lemma 7]), i.e.,

$$\lim_{n\to\infty}\left\|\hat{x}^n-\hat{x}^{n+1}\right\|_{\ell_2}=0,\quad \lim_{n\to\infty}\left\|\tilde{x}^n-\tilde{x}^{n+1}\right\|_{\ell_2}=0,\quad \text{and}\quad \lim_{n\to\infty}\left\|\hat{x}^n-\tilde{x}^n\right\|_{\ell_2}=0. \tag{86}$$

Since $0\leq\varepsilon^{n+1}\leq\varepsilon^n$ the ε^n always converge to some $\varepsilon\geq 0$.

Case $\varepsilon=0$: By means of the boundedness of the sequence $\tilde{x}^n$, and the definition of ε^n, we show that there is a subsequence $(\tilde{x}^{p_j})_{p_j\in\mathbb{N}}$ of $(\tilde{x}^n)_{n\in\mathbb{N}}$ such that $\tilde{x}^{p_j}\to\bar{x}\in\mathcal{F}_\Phi(y)$ and $\bar{x}$ is the unique ℓ_1-minimizer. We still need to show that also $\tilde{x}^n\to\bar{x}$. For that, we notice first that $\tilde{x}^{p_j}\to\bar{x}$ and $\varepsilon^{p_j}\to 0$ imply $J\left(\tilde{x}^{p_j},w^{p_j},\varepsilon^{p_j}\right)\to\|\bar{x}\|_{\ell_1}$. The convergence of $J\left(\tilde{x}^n,w^n,\varepsilon^n\right)\to\|\bar{x}\|_{\ell_1}$ is established by the following argument: For each $n\in\mathbb{N}$ there is exactly one i such that $p_i<n\leq p_{i+1}$. We can construct the telescopic sum

$$J\left(\tilde{x}^n,w^n,\varepsilon^n\right)-J\left(\tilde{x}^{p_i},w^{p_i},\varepsilon^{p_i}\right)=\sum_{k=p_i}^{n-1}J\left(\tilde{x}^{k+1},w^{k+1},\varepsilon^{k+1}\right)-J\left(\tilde{x}^{k},w^{k},\varepsilon^{k}\right),$$

and estimate by means of (85) and (83)

$$\sum_{k=p_i}^{n-1}J\left(\tilde{x}^{k+1},w^{k+1},\varepsilon^{k+1}\right)-J\left(\tilde{x}^{k},w^{k},\varepsilon^{k}\right)\leq 4\sum_{k=p_i}^{n-1}a_{k+1}\leq 4\left(A-\sum_{k=1}^{p_i-1}a_{k+1}\right),$$

for $A:=\sum_{k=0}^{\infty}a_k$. Thus, letting $n\to\infty$ implies $i(n)\to\infty$ and leads to

$$\limsup_{n\to\infty}J\left(\tilde{x}^{p_i},w^{p_i},\varepsilon^{p_i}\right)-J\left(\tilde{x}^{n},w^{n},\varepsilon^{n}\right)\leq 4\left(A-\sum_{k=1}^{\infty}a_{k+1}\right)=0,$$

and therefore

$$\limsup_{n\to\infty}J\left(\tilde{x}^n,w^n,\varepsilon^n\right)\leq\|\bar{x}\|_{\ell_1}.$$

A similar argument can be done considering the difference $J\left(\tilde{x}^{p_{i+1}},w^{p_{i+1}},\varepsilon^{p_{i+1}}\right)-J\left(\tilde{x}^{n},w^{n},\varepsilon^{n}\right)$ instead, yielding also to the converse estimate

$$\limsup_{n\to\infty}J\left(\tilde{x}^n,w^n,\varepsilon^n\right)\geq\|\bar{x}\|_{\ell_1}.$$

Eventually, one concludes the convergence of the limit, by substituting in the previous argument the "lim sup" with the "lim inf". Moreover (42) implies

$$J(\tilde{x}^n, w^n, \varepsilon^n) - N(\varepsilon^n) \leq \|\tilde{x}^n\|_{\ell_1} \leq J(\tilde{x}^n, w^n, \varepsilon^n),$$

and thus, we obtain $\|\tilde{x}^n\|_{\ell_1} \to \|\bar{x}\|_{\ell_1}$. Finally we invoke Lemma [22, Lemma 4.2] with $z' = \tilde{x}^n$ and $z = \bar{x}$ to get

$$\limsup_{n\to\infty} \|\tilde{x}^n - \bar{x}\|_{\ell_1} \leq \frac{1+\gamma}{1-\gamma}\left(\lim_{n\to\infty} \|\tilde{x}^n\|_{\ell_1} - \|\bar{x}\|_{\ell_1}\right) = 0,$$

which completes the proof of $\tilde{x}^n \to \bar{x}$ in this case.
To see (74) and establish (i), invoke [22, Lemma 4.3] and use the fact that $\sigma_k(x)_{\ell_1} \geq \sigma_K(x)_{\ell_1}$ for $k \leq K$.

Case $\varepsilon > 0$: Again we use the fact that $(\tilde{x}^n)_{n\in\mathbb{N}}$ is a bounded sequence and hence has accumulation points. Let $(\tilde{x}^{n_i})$ be any convergent subsequence of $(\tilde{x}^n)_{n\in\mathbb{N}}$ and let $\bar{x} \in \mathcal{Z}(y)$ its limit. By the fact that $\lim_{n\to\infty} \|\hat{x}^n - \tilde{x}^n\|_{\ell_2} = 0$ (compare (86)), we know that also $\bar{x} \in \mathcal{F}_\Phi(y)$. Following the steps in the proof of Theorem 6, one shows that $\langle \bar{x}, \eta \rangle_{\hat{w}(\bar{x},\varepsilon,\tau)} = 0$ for all $\eta \in \mathcal{N}_\Phi$. Thus, we have by means of Lemma 7 that $\bar{x} = x^\varepsilon$. Hence, x^ε is the unique accumulation point of $(\tilde{x}^n)_{n\in\mathbb{N}}$. This establishes (ii).

To prove (75), follow the steps in the proof of Theorem 6, and adapt it according to the ε-update rule with β instead of $\frac{1}{N}$, to conclude. □

Proof (Theorem 9). The proof is a minimal modification of the one of Theorem 7. Due to the fact that in general $\tilde{x}^n \neq \hat{x}^n$, the calculation slightly changes. The details can be found in [30, Section 3.3.4]. □

2.2.1.6 A surprising superlinear convergence promoting ℓ_τ-minimization for $\tau < 1$

The linear rate (65) can be improved significantly, by a very simple modification of the rule of updating the weight:

$$w_j^{n+1} = \left((x_j^{n+1})^2 + (\varepsilon^{n+1})^2\right)^{-\frac{2-\tau}{2}}, \quad j = 1, \ldots, N, \text{ for any } 0 < \tau < 1.$$

This corresponds to the substitution of the function J with

$$J_\tau(z, w, \varepsilon) := \frac{\tau}{2}\left[\sum_{j=1}^N z_j^2 w_j + \sum_{j=1}^N \left(\varepsilon^2 w_j + \frac{2-\tau}{\tau} w_j^{-\frac{\tau}{2-\tau}}\right)\right],$$
$$z \in \mathbb{R}^N, w \in \mathbb{R}_+^N, \varepsilon \in \mathbb{R}_+.$$

Surprisingly the rate of local convergence of this modified algorithm is superlinear; the rate is larger for smaller τ, increasing to approach a quadratic regime as $\tau \to 0$. More precisely the local error $E_n := \|x^n - x^*\|_{\ell_\tau^N}^\tau$ satisfies

$$E_{n+1} \leqslant \mu(\gamma, \tau) E_n^{2-\tau}, \tag{87}$$

where $\mu(\gamma, \tau) < 1$ for $\gamma > 0$ sufficiently small. The validity of (87) is restricted to x^n in a (small) ball centered at x^*. In particular, if x^0 is close enough to x^* then (87) ensures the convergence of the algorithm to the k-sparse solution x^*. We refer the reader to [22, 30] for more details. Note that these error estimates are sharp for the non-accelerated IRLS method and one is able to see these effects in numerical simulations. Maybe the most surprising result is that the Conjugate Gradient accelerated IRLS cannot significantly be accelerated by means of decreasing τ. It turns out that the latter modification in practice makes the algorithm more unstable but not necessarily leads to a faster convergence. In Section 4, we show these effects in a numerical simulation.

2.2.2 IRLS method for ℓ_1-norm regularization

In the previous chapter the solution x^* was intended to solve the linear system $\Phi x = y$ exactly. In most engineering and physical applications such a setting is not applicable since the measurements are perturbed by noise. Therefore, consider problem (26) and the functional $\mathcal{J}_\lambda(x)$ where the parameter λ balances the residual error in the linear system with an ℓ_1-norm penalty, promoting sparsity.

2.2.2.1 The algorithm IRLS-λ

Definition 3. Given a real number $\varepsilon > 0$, $x \in \mathbb{R}^N$, and a weight vector $w \in \mathbb{R}^N$, $w > 0$, we define

$$J_\lambda(x,w,\varepsilon) := \frac{1}{2}\sum_{j=1}^{N}\left[|x_j|^2 w_j + \varepsilon^2 w_j + w_j^{-1}\right] + \frac{1}{2\lambda}||\Phi x - y||_{\ell_2}^2. \tag{88}$$

Lai, Xu, and Yin in [46] and Voronin in [75] showed independently that problem (26) can be approached by an alternating minimization of the functional J_λ with respect to x, w, and ε. The difference between these two works is the definition of the update rule for ε. Here, we chose the rule proposed by Voronin, and present Algorithm IRLS-λ. The choice of this particular update rule is justified by the reason that it allows us not only to show that the algorithm is converging to a minimizer of (26), but the concept can be easily extended to find critical points of the ℓ_τ-regularized problem

$$\min_x \left(\mathcal{J}_{\tau,\lambda}(x) := ||x||_{\ell_\tau}^\tau + \frac{1}{2\lambda}||\Phi x - y||_{\ell_2}^2\right), \tag{89}$$

for $0 < \tau \leq 1$, also for the case where we wish to accelerate this method by CG. However, we could not prove similar statements by means of the rule of Lai, Xu, and Yin, which only allows to show the convergence of the algorithm to a critical point of the smoothed functional

$$\min_x ||x||_{\ell_\tau,\varepsilon}^\tau + \frac{1}{2\lambda}||\Phi x - y||_{\ell_2}^2,$$

where $||x||_{\ell_\tau,\varepsilon}^\tau := \sum_{j=1}^{N} |x_j^2 + \varepsilon^2|^{\frac{\tau}{2}}$, where $\varepsilon := \lim_{n\to\infty}\varepsilon^n$. For the sake of simplicity and consistency, we restrict ourselves to the convex case and problem (26). A short comment on the non-convex version is given at the end of this section. For more precise statements we refer the reader to [30].

Algorithm 3 IRLS-λ

1: Set $w^0 := (1,\dots,1)$, $\varepsilon^0 := 1$, $\alpha \in (0,1]$, $\phi \in (0, \frac{1}{3})$.
2: **while** $\varepsilon^n > 0$ **do**
3: $\quad x^{n+1} := \arg\min_x J_\lambda(x, w^n, \varepsilon^n)$
4: $\quad \varepsilon^{n+1} := \min\left\{\varepsilon^n, |J_\lambda(\tilde{x}^{n-1}, w^{n-1}, \varepsilon^{n-1}) - J_\lambda(\tilde{x}^n, w^n, \varepsilon^n)|^\phi + \alpha^{n+1}\right\}$
5: $\quad w^{n+1} := \arg\min_{w>0} J_\lambda(x^{n+1}, w, \varepsilon^{n+1})$
6: **end while**

We actually approach the first step of the algorithm by means of obtaining a critical point of $J_\lambda(\cdot, w, \varepsilon)$ by the first order optimality condition

$$\left[x_j w_j^n\right]_{j=1,\dots,N} + \frac{1}{\lambda}\Phi^*(\Phi x - y) = 0, \tag{90}$$

or equivalently by solving the linear system

$$\left(\Phi^*\Phi + \operatorname{diag}\left[\lambda w_j^n\right]_{j=1}^N\right) x = \Phi^* y. \tag{91}$$

We denote the solution of this system by x^{n+1}. The new weight w^{n+1} is obtained in step 3 and can be expressed componentwise by

$$w_j^{n+1} = ((x_j^{n+1})^2 + (\varepsilon^{n+1})^2)^{-\frac{1}{2}}. \tag{92}$$

2.2.2.2 Conjugate gradient acceleration of IRLS method for ℓ_1-norm regularization

In the following we again propose the combination of Algorithm IRLS with the CG method (see Appendix 8). CG is used to calculate an approximation of the solution of the linear system (91) in step 3 of the algorithm. After including the CG method, the modified algorithm which we shall consider is Algorithm CG-IRLS-λ.

Algorithm 4 CG-IRLS-λ

1: Set $w^0 := (1,\dots,1)$, $\varepsilon^0 := 1$, $\alpha \in (0,1]$, $\phi \in (0, \frac{1}{3})$.
2: **while** $\varepsilon^n > 0$ **do**
3: Compute $\tilde{x}^{n+1}$ by means of CG, s.t. $||\tilde{x}^{n+1} - \hat{x}^{n+1}||_{\ell_2(w^n)} \le \mathrm{tol}_{n+1}$,
where $\hat{x}^{n+1} := \arg\min_x J_\lambda(x, w^n, \varepsilon^n)$
4: $\varepsilon^{n+1} := \min\left\{\varepsilon^n, |J_\lambda(\tilde{x}^{n-2}, w^{n-2}, \varepsilon^{n-2}) - J_\lambda(\tilde{x}^{n-1}, w^{n-1}, \varepsilon^{n-1})|^\phi + \alpha^{n+1}\right\}$
5: $w^{n+1} := \arg\min_{w>0} J_\lambda(\tilde{x}^{n+1}, w, \varepsilon^{n+1})$
6: **end while**

Notice that $\tilde{x}$ always denotes the approximate solution of the minimization with respect to x in step 3 and $\hat{x}$ the corresponding exact solution. Thus $\hat{x}^{n+1}$ fulfills (91) but not $\tilde{x}^{n+1}$.

In Theorem 21 a stopping condition for the CG method is provided, it is again not practical for us, since we do not dispose of the minimizer and the computation of the condition number is computationally expensive. Therefore, we provide a stopping criterion to make sure that $||\tilde{x}^{n+1} - \hat{x}^{n+1}||_{\ell_2(w^n)} \le \mathrm{tol}_{n+1}$ is fulfilled in step 3 of Algorithm CG-IRLS-λ.

Let $\tilde{x}^{n+1,l}$ be the l-th iterate of the CG method and define

$$A_n := \Phi^*\Phi + \operatorname{diag}\left[\lambda w_j^n\right]_{j=1}^N.$$

Notice that the matrix $\Phi^*\Phi$ is positive semi-definite and $\lambda D_n^{-1} = \lambda \operatorname{diag}\left[w_j^n\right]_{j=1}^N$ is positive definite. Therefore, A_n is positive definite and invertible, and furthermore we have the estimate for the smallest eigenvalue

$$\lambda_{\min}(A_n) \ge \lambda_{\min}(\operatorname{diag}\left[\lambda w_j^n\right]_{j=1}^N). \tag{93}$$

We obtain

$$\left\|\hat{x}^{n+1}-\tilde{x}^{n+1,l}\right\|_{\ell_2(w^n)} \leq \left\|A_n^{-1}\left(\Phi^* y - A_n\tilde{x}^{n+1,l}\right)\right\|_{\ell_2(w^n)} \leq \left\|D_n^{-\frac{1}{2}}\right\| \left\|A_n^{-1}\right\| \left\|r^{n+1,l}\right\|_{\ell_2}, \tag{94}$$

where $r^{n+1,l} := \Phi^* y - A_n\tilde{x}^{n+1,l}$ is the residual as it appears in step 5 of Algorithm 8. The first factor on the right-hand side of (94) can be estimated by

$$\left\|D_n^{-\frac{1}{2}}\right\| = \lambda_{\max}\left(D_n^{-\frac{1}{2}}\right) = \sqrt{\max_j w_j^n} = \sqrt{\max_j \left(\left(\tilde{x}_j^n\right)^2 + (\varepsilon^n)^2\right)^{-\frac{1}{2}}} \leq (\varepsilon^n)^{-\frac{1}{2}}.$$

The second factor of (94) is estimated by

$$\left\|A_n^{-1}\right\| = (\lambda_{\min}(A_n))^{-1} \leq \left(\lambda_{\min}(\operatorname{diag}\left[\lambda w_j^n\right]_{j=1}^N)\right)^{-1} = \left(\lambda\left(\left(\max_j |\tilde{x}_j^n|\right)^2 + (\varepsilon^n)^2\right)^{-\frac{1}{2}}\right)^{-1},$$

where we used (93) in the inequality. Thus, we obtain

$$\left\|\tilde{x}^{n+1}-\hat{x}^{n+1,l}\right\|_{\ell_2(w^n)} \leq \frac{\left(\left(\max_j |\tilde{x}_j^n|\right)^2 + (\varepsilon^n)^2\right)^{\frac{1}{2}}}{(\varepsilon^n)^{\frac{1}{2}}\,\lambda} \left\|r^{n+1,l}\right\|_{\ell_2},$$

and the suitable stopping condition

$$\left\|r^{n+1,l}\right\|_{\ell_2} \leq \frac{(\varepsilon^n)^{\frac{1}{2}}\,\lambda}{\left(\left(\max_j |\tilde{x}_j^n|\right)^2 + (\varepsilon^n)^2\right)^{\frac{1}{2}}}\,\mathrm{tol}_{n+1}. \tag{95}$$

In the following Theorem we clarify how to choose the tolerance tol_{n+1} in order to establish a convergence result of the algorithm.

Theorem 10. *Let $\lambda > 0$, $\Phi \in \mathbb{R}^{m\times N}$, and $y \in \mathbb{R}^m$. Define the sequences $(\tilde{x}^n)_{n\in\mathbb{N}}$, $(\varepsilon^n)_{n\in\mathbb{N}}$ and $(w^n)_{n\in\mathbb{N}}$ as generated by Algorithm CG-IRLS-λ. Choose the accuracy tol_n of the CG-method, such that*

$$\mathrm{tol}_n \;\leq\; \min\left\{a_n\left(\sqrt{2\bar{J}}C_{w^{n-1}} + 2\sqrt{\frac{2}{\lambda}\bar{J}}||\Phi||\right)^{-1}, \sqrt{a_n}\left(\frac{1}{2} + \frac{||\Phi||^2}{2\lambda}\bar{J}\right)^{-\frac{1}{2}}\right\}, \tag{96}$$

$$\text{with } C_{w^{n-1}} := \left(\frac{\max_j(\tilde{x}_j^{n-1})^2 + (\varepsilon^{n-1})^2}{(\varepsilon^n)^2}\right)^{\frac{1}{2}}, \tag{97}$$

where a_n is chosen as a summable sequence, and $\bar{J} := J_\lambda(\tilde{x}^1, w^0, \varepsilon^0)$. Then the sequence $(\tilde{x}^n)_{n\in\mathbb{N}}$ has at least one convergent subsequence $(\tilde{x}^{n_k})_{k\in\mathbb{N}}$ with limit x^λ, which is a minimizer of $\mathcal{J}_\lambda(x)$, i.e.,

$$-(\Phi^*(y-\Phi x^\lambda))_j = \lambda \operatorname{sign}(x_j^\lambda), \quad x_j^\lambda \neq 0, \tag{98}$$

$$|(\Phi^*(y-\Phi x^\lambda))_j| \leq \lambda, \quad x_j^\lambda = 0, \tag{99}$$

if $x^\lambda \neq 0$.

Remark 2. Actually one can show that the limit of any convergent subsequence produced by Algorithm CG-IRLS-λ fulfills the conditions (98) and (99). This result appears in [75, Lemma 4.6.1] for IRLS-λ and can be adapted to CG-IRLS-λ, since it is based on the monotonicity of the functional J_λ, which is shown in [30].

Proof. This Theorem is a straightforward adaptation of [30, Theorem 5] for $\tau = 1$. In the respective proof this case is treated separately and can be followed at it is. □

2.2.2.3 Nonconvex ℓ_τ-norm Regularization

If $0 < \tau < 1$, the problem (89) is nonconvex and non-smooth. Necessary first order optimality conditions for a global minimizer of this functional were derived in [8, Proposition 3.14], and [42, Theorem 2.2]. In our case, we are able to show that the non-zero components of the limits of the algorithm fulfill the respective optimality conditions. However, as soon as the algorithm is producing zeros in some components of the limit, so far, we were not able to verify them directly. On this account, we pursue a different strategy, which originates from [76]. We do not directly show that the algorithm computes a solution of problem (89). Instead we show that a subsequence of the algorithm is at least computing a point $x^\dagger$, whose transformation $\check{x}^\dagger = \mathcal{N}_{\upsilon/\tau}^{-1}(x^\dagger)$ is a *critical point* of the functional

$$\check{\mathcal{J}}_{\upsilon,\lambda}(x) := \|x\|_{\ell_\upsilon}^\upsilon + \frac{1}{2\lambda}\left\|\Phi\mathcal{N}_{\upsilon/\tau}(x) - y\right\|_{\ell_2}^2, \tag{100}$$

where

$$\mathcal{N}_\zeta \colon \mathbb{R}^N \to \mathbb{R}^N, \quad \left(\mathcal{N}_\zeta(x)\right)_j := \operatorname{sign}(x_j)|x_j|^\zeta, \quad j = 1,\ldots,N, \tag{101}$$

is a continuous bijective mapping and $1 < \upsilon \leq 2$. It was shown in [64, 76] that assuming $\check{x}^\dagger$ is a global minimizer of $\check{\mathcal{J}}_{\upsilon,\lambda}(x)$ implies that $x^\dagger$ is also a global minimizer of $\mathcal{J}_{\tau,\lambda}$, i.e., a solution of problem (26).

In a slight adaptation of Theorem 10, which is stated in [30], it is shown that an alternating minimization of the functional

$$J_{\tau,\lambda}(x,w,\varepsilon) := \frac{\tau}{2}\sum_{j=1}^N \left[|x_j|^2 w_j + \varepsilon^2 w_j + \frac{2-\tau}{\tau} w_j^{-\frac{\tau}{2-\tau}}\right] + \frac{1}{2\lambda}||\Phi x - y||_{\ell_2}^2. \tag{102}$$

produces at least one convergent subsequence $(\tilde{x}^{n_k})_{k\in\mathbb{N}}$ with limit x^λ. The transformation of the limit $\check{x}^\lambda := \mathcal{N}_{\upsilon/\tau}^{-1}(x^\lambda)$, $1 < \upsilon \leq 2$, as defined in (101), is a critical point of (100). If $\check{x}^\lambda$ is a global minimizer of (100), then x^λ is also a global minimizer of $\mathcal{J}_{\tau,\lambda}(x)$.

This result can be partially extended towards local minimizers. For completeness of the presentation we sketch here the argument from [64]. Assume that $\check{x}^\lambda$ is a local minimizer. Then there is a neighborhood $U_\varepsilon(\check{x}^\lambda)$ with $\varepsilon > 0$ such that for all $x' \in U_\varepsilon(\check{x}^\lambda)$:

$$\check{\mathcal{J}}_{\upsilon,\lambda}(x') \geq \check{\mathcal{J}}_{\upsilon,\lambda}(\check{x}^\lambda).$$

By continuity of $\mathcal{N}_{\upsilon/\tau}$ there is an $\hat{\varepsilon} > 0$ such that the neighborhood $U_{\hat{\varepsilon}}(x^\lambda) \subset \mathcal{N}_{\upsilon/\tau}(U_\varepsilon(\check{x}^\lambda))$. Thus, for all $x \in U_{\hat{\varepsilon}}(x^\lambda)$, we have $x' = \mathcal{N}_{\upsilon/\tau}^{-1}(x) \in U_\varepsilon(\check{x}^\lambda)$, and obtain

$$\begin{aligned}\mathcal{J}_{\tau,\lambda}(x) &= ||x||_{\ell_\tau}^\tau + \frac{1}{2\lambda}||\Phi x - y||_{\ell_2}^2 = ||\mathcal{N}_{\upsilon/\tau}(x')||_{\ell_\tau}^\tau + \frac{1}{2\lambda}||\Phi\mathcal{N}_{\upsilon/\tau}(x') - y||_{\ell_2}^2 \\ &= ||x'||_{\ell_\upsilon}^\upsilon + \frac{1}{2\lambda}||\Phi\mathcal{N}_{\upsilon/\tau}(x') - y||_{\ell_2}^2 = \check{\mathcal{J}}_{\upsilon,\lambda}(x') \geq \check{\mathcal{J}}_{\upsilon,\lambda}(\check{x}^\lambda) = ||\check{x}^\lambda||_{\ell_\upsilon}^\upsilon + \frac{1}{2\lambda}||\Phi\mathcal{N}_{\upsilon/\tau}(\check{x}^\lambda) - y||_{\ell_2}^2 \\ &= ||x^\lambda||_{\ell_\tau}^\tau + \frac{1}{2\lambda}||\Phi x^\lambda - y||_{\ell_2}^2 = \mathcal{J}_{\tau,\lambda}(x^\lambda).\end{aligned}$$

2.3 Iterative Hard Thresholding (IHT)

Let us now return to the sparse recovery problem (7) and introduce the Iterative Hard Thresholding (IHT) algorithm. This algorithm can be seen as a method to solve (27) for a suitable $\lambda = \lambda(k) > 0$, or equivalently for the solution of the optimization problem (25). Both problems (27) and (25) are solved by iterating

$$x^{n+1} = \mathbb{H}(x^n + \Phi^*(y - \Phi x^n)), \tag{103}$$

where $\mathbb{H}$ is an hard thresholding operator, which depends on the respective problem formulation.

It turns out that in the case of problem (25) the thresholding operator $\mathbb{H}(z) = \mathbb{H}_k(z) := z_{[k]}$, which returns the best k-term approximation to z (see (3)), is appropriate. Note that if x^* is k-sparse and $\Phi x^* = y$, then x^* is a fixed point of

$$x^* = \mathbb{H}_k(x^* + \Phi^*(y - \Phi x^*)).$$

In Section 2.3.1, we present a proper analysis of this algorithm and show that under the RIP for Φ, it converges to a local minimizer of (25) and has stability properties as in (22), which are reached in a finite number of iterations.

In order to solve problem (27), the thresholding operator is given by $\mathbb{H}(z) := \mathbb{H}_{\sqrt{\lambda}}(z)$ where

$$(\mathbb{H}_{\sqrt{\lambda}}(z))_i := \begin{cases} z_i & \text{if } |z_i| > \sqrt{\lambda}, \\ 0 & \text{else,} \end{cases}, \ i = 1\ldots,N.$$

In Section 2.3.2, we present a theorem that states the convergence of the algorithm to a fixed point x^h fulfilling

$$x^\mathrm{h} := \mathbb{H}_{\sqrt{\lambda}}\left(x^\mathrm{h} + \Phi^*(y - \Phi x^\mathrm{h})\right), \tag{104}$$

which is a local minimizer of the functional (27).

2.3.1 IHT for the ℓ_0-constrained Problem

Algorithm 5 IHT-k

1: Set $x^0 := 0$.
2: **loop**
3: $z^{n+1} := x^n + \Phi^*(y - \Phi x^n)$
4: $x^{n+1} := \mathbb{H}_k(z) = z_{[k]}$
5: $n := n+1$
6: **end loop**

We first specify in Algorithm IHT-k the formulation of IHT for problem (25), which was only roughly described in the preceding introduction. It was shown in [6] that if $\|\Phi\| < 1$ then this algorithm converges to a local minimizer of (27). We would like to analyze this algorithm following [7] in the case Φ satisfies the RIP and establish a first convergence result.

Theorem 11. *Let us assume that $y = \Phi x + e$ is a noisy encoding of x via Φ, where x is k-sparse. If Φ has the RIP of order $3k$ and constant $\delta_{3k} < \frac{1}{12\sqrt{2}}$, then, at iteration n, Algorithm IHT-k will recover an approximation x^n satisfying*

$$\|x - x^n\|_{\ell_2} \leq 2^{-n}\|x\|_{\ell_2} + 5\|e\|_{\ell_2}. \tag{105}$$

Furthermore, after at most

$$n^* = \left\lceil \log_2 \left(\frac{\|x\|_{\ell_2}}{\|e\|_{\ell_2}} \right) \right\rceil \tag{106}$$

iterations, the algorithm estimates x with accuracy

$$\|x - x^{n^*}\|_{\ell_2} \leq 6\|e\|_{\ell_2}. \tag{107}$$

Proof. Let us denote $z^n := x^n + \Phi^*(y - \Phi x^n)$, $r^n = x - x^n$, and $\Lambda^n := \operatorname{supp}(r^n)$. By triangle inequality we can write

$$\|x - x^{n+1}\|_{\ell_2} = \|(x - x^{n+1})_{\Lambda^{(n+1)}}\|_{\ell_2} \leq \|x_{\Lambda^{n+1}} - z^n_{\Lambda^{n+1}}\|_{\ell_2} + \|x^{n+1}_{\Lambda^{n+1}} - z^n_{\Lambda^{n+1}}\|_{\ell_2}.$$

By definition $x^{n+1} = \mathbb{H}_k(z^n) = z^{(n)}_{[k]}$. This implies

$$\|x^{n+1}_{\Lambda^{n+1}} - z^n_{\Lambda^{n+1}}\|_{\ell_2} \leq \|x_{\Lambda^{n+1}} - z^n_{\Lambda^{n+1}}\|_{\ell_2},$$

and

$$\|x - x^{n+1}\|_{\ell_2} \leq 2\|x_{\Lambda^{n+1}} - z^n_{\Lambda^{n+1}}\|_{\ell_2}.$$

We can also write

$$z^n_{\Lambda^{n+1}} = x^n_{\Lambda^{n+1}} + \Phi^*_{\Lambda^{n+1}}\Phi r^n + \Phi^*_{\Lambda^{n+1}} e.$$

We then have

$$\begin{aligned}\|x - x^{n+1}\|_{\ell_2} &\leq 2\|x_{\Lambda^{n+1}} - x^n_{\Lambda^{n+1}} - \Phi^*_{\Lambda^{n+1}}\Phi r^n - \Phi^*_{\Lambda^{n+1}} e\|_{\ell_2} \\ &\leq 2\|(I - \Phi^*_{\Lambda^{n+1}}\Phi_{\Lambda^{n+1}})r^n\|_{\ell_2} + 2\|\Phi^*_{\Lambda^{n+1}}\Phi_{\Lambda^n \setminus \Lambda^{n+1}} r^n\|_{\ell_2} + 2\|\Phi^*_{\Lambda^{n+1}} e\|_{\ell_2}.\end{aligned}$$

Note that $|\Lambda^n| \leq 2k$ and that $|\Lambda^{n+1} \cup \Lambda^n| \leq 3k$. By an application of the bounds in Lemma 2, and by using the fact that $\delta_{2k} \leq \delta_{3k}$ (note that a $2k$-sparse vector is also $3k$-sparse)

$$\|r^{n+1}\|_{\ell_2} \leq 6\delta_{2k}\|r^n_{\Lambda^{n+1}}\|_{\ell_2} + 6\delta_{3k}\|r^n_{\Lambda^n \setminus \Lambda^{n+1}}\|_{\ell_2} + 2(1 + \delta_{2k})\|e\|_{\ell_2}$$

Moreover $\|r^n_{\Lambda^{n+1}}\|_{\ell_2} + \|r^n_{\Lambda^n \setminus \Lambda^{n+1}}\|_{\ell_2} \leq \sqrt{2}\|r^n\|_{\ell_2}$. Therefore we have the bound

$$\|r^{n+1}\|_{\ell_2} \leq 6\sqrt{2}\delta_{3k}\|r^n\|_{\ell_2} + 2(1+\delta_{3k})\|e\|_{\ell_2}.$$

By assumption $\delta_{3k} < \frac{1}{12\sqrt{2}}$ and $6\sqrt{2}\delta_{3k} < \frac{1}{2}$. (Note that here we could simply choose any value $\delta_{3k} < \frac{1}{6\sqrt{2}}$ and obtain a slightly different estimate!) Then we get the recursion

$$\|r^{n+1}\|_{\ell_2} \leq 2^{-1}\|r^n\|_{\ell_2} + 2.12\|e\|_{\ell_2},$$

which iterated (note that $x^0 = 0$ and $2.12\sum_{n=0}^{\infty} 2^{-n} \leq 4.24$) gives

$$\|r^{n+1}\|_{\ell_2} \leq 2^{-n}\|x\|_{\ell_2} + 4.24\|e\|_{\ell_2}.$$

This is precisely the bound we were looking for. The rest of the statements of the theorem is left as an exercise. □

We have also the following result.

Corollary 1. *Let us assume that $y = \Phi x + e$ is a noisy encoding of x via Φ, where x is* an arbitrary vector. *If Φ has the RIP of order $3k$ and constant $\delta_{3k} < \frac{1}{6\sqrt{2}}$, then, at iteration n, Algorithm IHT-k will recover an approximation x^n satisfying*

$$\|x - x^n\|_{\ell_2} \leq 2^{-n}\|x\|_{\ell_2} + 6\left(\sigma_k(x)_{\ell_2} + \frac{\sigma_k(x)_{\ell_1}}{\sqrt{k}} + \|e\|_{\ell_2}\right). \tag{108}$$

Furthermore, after at most

$$n^* = \left\lceil \log_2\left(\frac{\|x\|_{\ell_2}}{\|e\|_{\ell_2}}\right)\right\rceil \tag{109}$$

iterations, the algorithm estimates x with accuracy

$$\|x - x^{n^*}\|_{\ell_2} \leq 7\left(\sigma_k(x)_{\ell_2} + \frac{\sigma_k(x)_{\ell_1}}{\sqrt{k}} + \|e\|_{\ell_2}\right). \tag{110}$$

Proof. We first note

$$\|x - x^n\|_{\ell_2} \leq \sigma_k(x)_{\ell_2} + \|x_{[k]} - x^n\|_{\ell_2}.$$

The proof now follows by bounding $\|x_{[k]} - x^n\|_{\ell_2}$. For this we simply apply Theorem 11 to $x_{[k]}$ with $\tilde{e}$ instead of e, and use Lemma 4 to bound $\|\tilde{e}\|_{\ell_2}$. The rest is left as a relatively simple derivation for the reader. □

2.3.2 IHT for the ℓ_0-regularized Problem

The rough formulation in the introduction is specified in Algorithm IHT-λ in order to solve (27). In the following, we present the convergence result of [6].

Theorem 12 ([6, Theorem 3, Lemma 4]). *If $\|\Phi\|_{\ell_2} < 1$, then the sequence $(x^n)_{n\in\mathbb{N}}$ defined by Algorithm 6 converges to a fixed point x^h of* (103)*, which is a local minimum of $\mathcal{J}_0(x)$. If further the set of columns $\{\Phi_i\}_{i=1}^N$ contains a basis for the signal space and $\|\Phi_i\|_{\ell_2} > 0$, $i = 1,\ldots,N$, then a tight bound for the approximation error at the fixed point x^h is*

Algorithm 6 IHT-λ

```
Set x^0 := 0.
loop
    z^{n+1} := x^n + Φ*(y − Φx^n)
    for i = 1,...,N do
        if z_i^{n+1} > √λ then
            x_i^{n+1} := z_i^{n+1}
        else
            x_i^{n+1} := 0
        end if
    end for
    n := n + 1
end loop
```

$$\|y-\Phi x^h\|_{\ell_2} \leq \frac{\sqrt{\lambda}}{\beta(\Phi)},$$

where $\beta(\Phi)>0$ *is such that*

$$\|\Phi^* z\|_{\ell_\infty} \geq \beta(\Phi)\|z\|_{\ell_2} \tag{111}$$

holds for all $z \in \mathbb{R}^m$.

Proof. Define the sets $\Lambda_0(z) := \{i : y_i = 0\}$ and $\Lambda_1(z) := \{i : |y_i| > \sqrt{\lambda}\}$. The proof of convergence is based on the fact that after a finite number of iterations these two sets are fixed. Thus the algorithm then can be considered as a standard Landweber algorithm with guaranteed convergence [47]. In the proof of Theorem 3 in [6] a detailed presentation of this argumentation, and the proof that the limit is a fixed point of (103) and therefore also a local minimum of $\mathcal{J}_0(x)$, is given. To show the approximation error estimate, we assume that the algorithm converged to a fixed point x^{h} which then has to fulfill (104). Since $\mathbb{H}_{\sqrt{\lambda}}$ is defined component-wise, we conclude that $|\Phi_i^*(y-\Phi x^{\mathrm{h}})| \leq \sqrt{\lambda}$ if $i \in \Lambda_0(x^{\mathrm{h}})$, and $\Phi_i^*(y-\Phi x^{\mathrm{h}}) = 0$ if $i \in \Lambda_1(x^{\mathrm{h}})$. Thus, we have in particular that $\|\Phi_i^*(y-\Phi x^{\mathrm{h}})\|_{\ell_\infty} \leq \sqrt{\lambda}$. By means of this observation and the application of condition (111) for $z = y-\Phi x^{\mathrm{h}}$, we obtain

$$\beta(\Phi)\|y-\Phi x^{\mathrm{h}}\|_{\ell_2} \leq \|\Phi^*(y-\Phi x^{\mathrm{h}})\|_{\ell_\infty} \leq \sqrt{\lambda}. \quad \square$$

Remark 3. Although it is the scope of the algorithm to produce a vector with small ℓ_0-norm - and thus a sparse vector - it is important to notice that this algorithm is only computing a local minimizer of the functional $\mathcal{J}_0(x)$, which is not necessarily sparse. In contrast to Algorithm IHT-k there is no guarantee, that this Algorithm produces a k-sparse vector.

Remark 4. In the proof, it is shown that the algorithm can be considered as a standard Landweber algorithm as soon as the sets $\Lambda_0(x^n)$ and $\Lambda_1(x^n)$ are fixed after a finite number of iterations n_0. According to [47], then the algorithm converges linearly as

$$\|x^n - x^{\mathrm{h}}\|_{\ell_2} \leq \|I - \Phi^*_{\Lambda_1(x^{n_0})}\Phi_{\Lambda_1(x^{n_0})}\|^{n-n_0}\|x^{n_0} - x^{\mathrm{h}}\|_{\ell_2}.$$

2.3.3 A brief comparison

At first glance algorithm IHT-k should be preferred to IHT-λ since it offers a more robust error analysis and a guaranteed error reduction from the very beginning and it is robust to noise, i.e., an estimate of the type (22) holds. However its main drawback is that it requires the (precise) knowledge of k, which one might not dispose of in some applications. Therefore in this case one can consider to use algorithm IHT-λ. Nevertheless one has to tune λ then. In Section 5 we present an application of the latter algorithm which turns out to be very robust when one is interested in the exact support identification of an original signal which is corrupted by noise. In this particular application we determine a specific λ which is supposed to provide optimal support identification performance.

The complexity of IHT algorithms is mainly depending on the application of $\Phi^*\Phi$. Thus, one would expect that they are greatly superior with respect to IRLS; however, we have to stress that IRLS can converge superlinearly and can be accelerated by means of the CG method, as we showed in Section 2.2. In Section 4 we present numerical tests which reveal that a CG-modified IRLS can be competitive with IHT in terms of computational complexity.

3 Numerical Methods for Sparse Recovery

In the previous chapters we put most of the emphasis on finite dimensional linear problems (also of relatively small size) where the model matrix Φ has the RIP or the NSP. This setting is suitable for applications in coding/decoding or compressed acquisition problems, hence from human-made problems coming from technology; however it does not fit many possible applications where we are interested in recovering quantities from partial real-life measurements. In this case we may need to work with large dimensional problems (even infinite dimensional) where the model linear (or nonlinear) operator which defines the measurements has not such nice properties as the RIP and NSP.

Here and later we are concerned with the more general setting and the efficient minimization of functionals of the type:

$$\mathcal{J}(x) := \|Kx-y\|_Y^2 + 2\|(\langle x, \tilde{\psi}_\alpha\rangle)_{\alpha\in\mathcal{I}}\|_{\ell_{1,\lambda}(\mathcal{I})}, \tag{112}$$

where $K : X \to Y$ is a bounded linear operator acting between two separable Hilbert spaces X and Y, $y \in Y$ is a given measurement, and $\Psi := (\psi_\alpha)_{\alpha\in\mathcal{I}}$ is a prescribed countable basis for X with associated dual $\tilde{\Psi} := (\tilde{\psi}_\alpha)_{\alpha\in\mathcal{I}}$. For $1 \le p < \infty$, the sequence norm $\|\mathbf{x}\|_{\ell_{p,\lambda}(\mathcal{I})} := (\sum_{\alpha\in\mathcal{I}} |u_\alpha|^p \lambda_\alpha)^{1/p}$ is the usual norm for weighted p-summable sequences, with weight $\lambda = (\lambda_\alpha)_{\alpha\in\mathcal{I}} \in \mathbb{R}_+^{\mathcal{I}}$, such that $\lambda_\alpha \ge \bar{\lambda} > 0$. To be concise with the meaning of the notation and help the reader to correctly associate the nature of the used veriables, in this section the notation of finite dimensional variables (e.g. x) is overloaded by the respective infinite dimensional meaning. Associated to the basis, we are given the synthesis map $F : \ell_2(\mathcal{I}) \to X$ defined by

$$F\mathbf{x} := \sum_{\alpha\in\mathcal{I}} x_\alpha \psi_\alpha, \quad \mathbf{x} \in \ell_2(\mathcal{I}). \tag{113}$$

We can re-formulate equivalently the functional in terms of sequences in $\ell_2(\mathcal{I})$ as follows:

$$\mathcal{J}(\mathbf{x}) := \mathcal{J}_\lambda(\mathbf{x}) = \|(K\circ F)\mathbf{x}-y\|_Y^2 + 2\|\mathbf{x}\|_{\ell_{1,\lambda}(\mathcal{I})}. \tag{114}$$

For ease of notation let us write $\Phi := K \circ F$. Such functional turns out to be very useful in many practical problems, where one cannot observe directly the quantities of most interest; instead their values have to be inferred from their effect on observable quantities. When this relationship between the observable y and the interesting quantity $\mathbf{x}$ is (approximately) linear the situation can be modeled mathematically by the equation

$$y = \Phi \mathbf{x} \,, \tag{115}$$

If Φ is a "nice" (e.g., well-conditioned), easily invertible operator, and if the data y are free of noise, then this is a well-known task which can be addressed with standard numerical analysis methods. Often, however, the mapping Φ is not invertible or ill-conditioned. Moreover, typically (115) is only an idealized version in which noise has been neglected; a more accurate model is

$$y = \Phi \mathbf{x} + e \,, \tag{116}$$

in which the data are corrupted by an (unknown) noise e. In order to deal with this type of reconstruction problem a *regularization* mechanism is required [27]. Regularization techniques try, as much as possible, to take advantage of (often vague) prior knowledge one may have about the nature of $\mathbf{x}$, which is embedded into the model. The approach modelled by the functional $\mathcal{J}$ in (112) is indeed tailored to the case when $\mathbf{x}$ can be represented by a *sparse* expansion, i.e., when x can be represented by a series expansion of the type (113) with respect to the basis Ψ (or a frame [20]) that has only a small number of large coefficients. The previous chapters should convince the reader that imposing an additional ℓ_1-norm term as in (113) has indeed the effect of sparsifying possible solutions. Hence, we model the sparsity constraint by a regularizing ℓ_1-term in the functional to be minimized; of course, we could consider also a minimization of the type (27), but that has the disadvantage of being nonconvex and not being necessarily robust to noise, when no RIP conditions are imposed on the model operator Φ. Naturally all the results of this section are also transferable to problem (26), which is the finite dimensional specialization of the above model with $\lambda_\alpha := \lambda$, $\alpha \in \mathcal{I}$.

In the following we will not use anymore the bold form $\mathbf{x}$ for a sequence in $\ell_2(\mathcal{I})$, since here and later we will exclusively work with the space $\ell_2(\mathcal{I})$.

3.1 Iterative Soft-Thresholding in Hilbert Spaces

Several authors have proposed independently an iterative soft-thresholding algorithm to approximate minimizers $x^* := x^*_\lambda$ of the functional in (113), see [25,28,67,68]. More precisely, x^* is the limit of sequences $x^{(n)}$ as defined in Algorithm ISTA, where $\mathbb{S}_\lambda$ is the soft-thresholding operation defined by $(\mathbb{S}_\lambda(x))_\alpha = S_{\lambda_\alpha}(x_\alpha)$ where

$$S_\tau(t) = \begin{cases} t-\tau & t > \tau \\ 0 & |t| \le \tau \\ t+\tau & t < -\tau \end{cases} . \tag{117}$$

Strong convergence of this algorithm was proved in [21], under the assumption that $\|\Phi\| < 1$ (actually, convergence can be shown also for $\|\Phi\| < \sqrt{2}$ [18]; nevertheless, the condition $\|\Phi\| < 1$ is by no means a restriction, since it can always be met by a suitable rescaling of the functional $\mathcal{J}$, in particular of K, y, and λ). Soft-thresholding plays a role in this problem because it leads to the unique minimizer of a functional combining ℓ_2 and ℓ_1-norms, i.e., (see Lemma 8 below)

Algorithm 7 Iterative Soft Thresholding Algorithm (ISTA)

Set $x^{(0)} \in \ell_2(\mathcal{I})$, for instance $x^{(0)} = 0$.

1: **loop**

2: $x^{(n+1)} = \mathbb{S}_\lambda\left(x^{(n)} + \Phi^* y - \Phi^*\Phi x^{(n)}\right)$

3: **end loop**

$$\mathbb{S}_\lambda(a) = \arg\min_{x\in\ell_2(\mathcal{I})}\left(\|x-a\|^2_{\ell_2(\mathcal{I})} + 2\|x\|_{\ell_{1,\lambda}(\mathcal{I})}\right). \tag{118}$$

We will call the iteration in step 2 of Algorithm ISTA the *iterative soft-thresholding algorithm* or the *thresholded Landweber iteration* (ISTA).

In this section we would like to provide the analysis of the convergence of this algorithm. Due to the lack of assumptions such as the RIP or the NSP, the methods we use come exclusively from convex analysis.

3.1.1 The Surrogate Functional

The first relevant observation is that the algorithm can be recast into an iterated minimization of a properly augmented functional, which we call the *surrogate functional* of $\mathcal{J}$, which is defined by

$$\mathcal{J}^S(x,a) := \|\Phi x - y\|^2_Y + 2\|x\|_{\ell_{1,\lambda}(\mathcal{I})} + \|x-a\|^2_{\ell_2(\mathcal{I})} - \|\Phi x - \Phi a\|^2_Y. \tag{119}$$

Assume here and later that $\|\Phi\| < 1$. Observe that, in this case, we have

$$\|x-a\|^2_{\ell_2(\mathcal{I})} - \|\Phi x - \Phi a\|^2_Y \geq C\|x-a\|^2_{\ell_2(\mathcal{I})}, \tag{120}$$

for $C = (1-\|\Phi\|^2) > 0$. Hence

$$\mathcal{J}(x) = \mathcal{J}^S(x,x) \leq \mathcal{J}^S(x,a), \tag{121}$$

and

$$\mathcal{J}^S(x,a) - \mathcal{J}^S(x,x) \geq C\|x-a\|^2_{\ell_2(\mathcal{I})}. \tag{122}$$

In particular, $\mathcal{J}^S$ is strictly convex with respect to x and it has a unique minimizer with respect to x once a is fixed. We have the following technical lemmas.

Lemma 8. *The soft-thresholding operator is the solution of the following optimization problem:*

$$\mathbb{S}_\lambda(a) = \arg\min_{x\in\ell_2(\mathcal{I})}\left(\|x-a\|^2_{\ell_2(\mathcal{I})} + 2\|x\|_{\ell_{1,\lambda}(\mathcal{I})}\right).$$

Proof. By componentwise optimization, we can reduce the problem to a scalar problem, i.e., we need to prove that

$$S_{\lambda_\alpha}(a_\alpha) = \arg\min_t (t-a_\alpha)^2 + 2\lambda_\alpha|t|,$$

which is shown by a simple direct computation. Let t^* be the minimizer. It is clear that $\operatorname{sign}(t^*)\operatorname{sign}(a_\alpha) \geq 0$ otherwise the function is increased. Hence we need to optimize $(t-a_\alpha)^2 + 2\lambda_\alpha \operatorname{sign}(a_\alpha)t$ which has

minimum at $\bar{t} = (a_\alpha - \operatorname{sign}(a_\alpha)\lambda_\alpha)$. If $|a_\alpha| > \lambda_\alpha$ than $t^* = \bar{t}$. Otherwise $\operatorname{sign}(\bar{t})\operatorname{sign}(a_\alpha) < 0$ and $\bar{t}$ cannot be the minimizer, and we have to choose $t^* = 0$. □

Lemma 9. *We can express the optimization of $\mathcal{J}^S(x,a)$ with respect to x explicitly by*

$$\mathbb{S}_\lambda(a + \Phi^*(y - \Phi a)) = \arg\min_{x \in \ell_2(\mathcal{I})} \mathcal{J}^S(x,a).$$

Proof. By developing the norm squares in (119) it is a straightforward computation to show

$$\mathcal{J}^S(x,a) = \|x - (a + \Phi^*(y - \Phi a))\|^2_{\ell_2(\mathcal{I})} + 2\|x\|_{\ell_{1,\lambda}(\mathcal{I})} + \Xi(a,A,y),$$

where $\Xi(a,A,y)$ is a function which does not depend on x. The statement follows now from an application of Lemma 8 and by the observation that the addition of constants to a functional does not modify its minimizer. □

By Lemma 9 we can replace $\mathbb{S}_\lambda\left(x^{(n)} + \Phi^* y - \Phi^*\Phi x^{(n)}\right) = \arg\min_{x \in \ell_2(\mathcal{I})} \mathcal{J}^S(x, x^{(n)})$ in step 2 of ISTA, which is a useful re-interpretation to conduct the remaining convergence analysis.

3.1.2 Preliminary Convergence Properties

Lemma 10. *The sequence $(\mathcal{J}(x^{(n)}))_{n\in\mathbb{N}}$ is nonincreasing. Moreover $(x^{(n)})_n$ is bounded in $\ell_2(\mathcal{I})$ and*

$$\lim_{n\to\infty} \|x^{(n+1)} - x^{(n)}\|^2_{\ell_2(\mathcal{I})} = 0. \tag{123}$$

Proof. Let us consider the estimates, which follow from the optimality of $x^{(n+1)}$, (121), and (122),

$$\mathcal{J}(x^{(n)}) = \mathcal{J}^S(x^{(n)}, x^{(n)}) \geq \mathcal{J}^S(x^{(n+1)}, x^{(n)}) \geq \mathcal{J}^S(x^{(n+1)}, x^{(n+1)}) = \mathcal{J}(x^{(n+1)}),$$

Hence, the sequence $\mathcal{J}(x^{(n)})$ is nonincreasing, and

$$\mathcal{J}(x^{(0)}) \geq \mathcal{J}(x^{(n)}) \geq 2\bar{\lambda}\|x^{(n)}\|_{\ell_1(\mathcal{I})} \geq 2\bar{\lambda}\|x^{(n)}\|_{\ell_2(\mathcal{I})}.$$

Therefore, $(x^{(n)})_n$ is bounded in $\ell_2(\mathcal{I})$. By (122), we have

$$\mathcal{J}(x^{(n)}) - \mathcal{J}(x^{(n+1)}) \geq C\|x^{(n)} - x^{(n+1)}\|^2_{\ell_2(\mathcal{I})}.$$

Since $\mathcal{J}(x^{(n)}) \geq 0$ is a nonincreasing sequence and is bounded below, it also converges, and

$$\lim_{n\to\infty} \|x^{(n+1)} - x^{(n)}\|^2_{\ell_2(\mathcal{I})} = 0.$$

□

As observed before for other algorithms, the convergence to zero of the difference of two iterates essentially means that the algorithm numerically converges. Moreover, by the uniform boundedness of $(x^{(n)})_{n\in\mathbb{N}}$, we know already that there are weakly converging subsequences. However, in order to conclude the convergence of the full sequence to a minimizer of $\mathcal{J}$ we need more technical work.

3.1.3 Weak Convergence of the Algorithm

As a simple exercise we state the following

Lemma 11. *The operator $\mathbb{S}_\lambda$ is nonexpansive, i.e.,*

$$\|\mathbb{S}_\lambda(x)-\mathbb{S}_\lambda(a)\|_{\ell_2(\mathcal{I})} \leq \|x-a\|_{\ell_2(\mathcal{I})}, \tag{124}$$

for all $x,a \in \ell_2(\mathcal{I})$.

Proof. Sketch: reason again componentwise and distinguish cases whether x_α and/or a_α are smaller or larger than the threshold $\pm\lambda_\alpha$. □

Moreover, we can characterize minimizers of $\mathcal{J}$ in the following way.

Proposition 1. *Define*

$$\Gamma(x) = \mathbb{S}_\lambda\left(x+\Phi^* y-\Phi^*\Phi x\right).$$

Then the set of minimizers of $\mathcal{J}$ coincides with the set $\mathrm{Fix}(\Gamma)$ of fixed points of Γ. In particular, since $\mathcal{J}$ is a coercive functional (i.e., $\{x : \mathcal{J}(x) \leq C\}$ is weakly compact for all $C > 0$), it has minimizers, and therefore Γ has fixed points.

Proof. Assume that x is the minimizer of $\mathcal{J}^S(\cdot,a)$, for a fixed. Let us now observe, first of all, that

$$\mathcal{J}^S(x+h,a) = \mathcal{J}^S(x,a) + 2\langle h, x-a-\Phi^*(y-\Phi a)\rangle + \sum_{\alpha\in\mathcal{I}} 2\lambda_\alpha(|x_\alpha+h_\alpha|-|x_\alpha|) + \|h\|^2_{\ell_2(\mathcal{I})}.$$

We define now $\mathcal{I}_0 = \{\alpha : x_\alpha = 0\}$ and $\mathcal{I}_1 = \mathcal{I}\setminus\mathcal{I}_0$. Since by Lemma 9 we have $x = \mathbb{S}_\lambda(a+\Phi^*(y-\Phi a))$, substituting it for x, we then have

$$\begin{aligned}\mathcal{J}^S(x+h,a) - \mathcal{J}^S(x,a) = \|h\|^2_{\ell_2(\mathcal{I})} &+ \sum_{\alpha\in\mathcal{I}_0}\left[2\lambda_\alpha|h_\alpha| - 2h_\alpha(a-\Phi^*(y-\Phi a))_\alpha\right] \\ &+ \sum_{\alpha\in\mathcal{I}_1}\left[2\lambda_\alpha|x_\alpha+h_\alpha| - 2\lambda_\alpha|x_\alpha| + h_\alpha(-2\lambda_\alpha\,\mathrm{sign}(x_\alpha))\right].\end{aligned}$$

If $\alpha\in\mathcal{I}_0$ then $|(a-\Phi^*(y-\Phi a))_\alpha| \leq \lambda_\alpha$, so that $2\lambda_\alpha|h_\alpha| - 2h_\alpha(a-\Phi^*(y-\Phi a))_\alpha \geq 0$. If $\alpha\in\mathcal{I}_1$, we distinguish two cases: if $x_\alpha > 0$, then

$$2\lambda_\alpha|x_\alpha+h_\alpha| - 2\lambda_\alpha|x_\alpha| + h_\alpha(-2\lambda_\alpha\,\mathrm{sign}(x_\alpha)) = 2\lambda_\alpha[|x_\alpha+h_\alpha| - (x_\alpha+h_\alpha)] \geq 0.$$

If $x_\alpha < 0$, then

$$2\lambda_\alpha|x_\alpha+h_\alpha| - 2\lambda_\alpha|x_\alpha| + h_\alpha(-2\lambda_\alpha\,\mathrm{sign}(x_\alpha)) = 2\lambda_\alpha[|x_\alpha+h_\alpha| + (x_\alpha+h_\alpha)] \geq 0.$$

It follows

$$\mathcal{J}^S(x+h,a) - \mathcal{J}^S(x,a) \geq \|h\|^2_{\ell_2(\mathcal{I})}. \tag{125}$$

Let us assume now that

$$x = \mathbb{S}_\lambda\left(x+\Phi^* y-\Phi^*\Phi x\right).$$

Then x is the minimizer of $\mathcal{J}^S(\cdot,x)$, and therefore

$$\mathcal{J}^S(x+h,x) \geq \mathcal{J}^S(x,x) + \|h\|^2_{\ell_2(\mathcal{I})}.$$

Observing now that $\mathcal{J}(x) = \mathcal{J}^S(x,x)$ and that $\mathcal{J}^S(x+h,x) = \mathcal{J}(x+h) + \|h\|^2_{\ell_2(\mathcal{I})} - \|\Phi h\|^2_Y$, we conclude that $\mathcal{J}(x+h) \geq \mathcal{J}(x) + \|\Phi h\|^2_Y$ for every h. Hence x is a minimizer of $\mathcal{J}$. Vice versa, if x is a minimizer of $\mathcal{J}$, then it is a minimizer of $\mathcal{J}^S(\cdot,x)$, and hence a fixed point of Γ. □

We need now to recall an important and well-known result related to iterations of nonexpansive maps [58]. We report it without proof; a simplified version of it can be also found in the Appendix B of [21].

Theorem 13 (Opial's Theorem). *Let a mapping Γ from $\ell_2(\mathcal{I})$ to itself satisfy the following conditions:*

(i) Γ is nonexpansive, i.e., $\|\Gamma(x) - \Gamma(a)\|_{\ell_2(\mathcal{I})} \leq \|x-a\|_{\ell_2(\mathcal{I})}$, for all $x, a \in \ell_2(\mathcal{I})$;
(ii) Γ is asymptotically regular, i.e., $\|\Gamma^{n+1}(x) - \Gamma^n(x)\|_{\ell_2(\mathcal{I})} \to 0$ for $n \to \infty$;
(iii) the set $\mathrm{Fix}(\Gamma)$ of its fixed points is not empty.

Then, for all x, the sequence $(\Gamma^n(x))_{n\in\mathbb{N}}$ converges weakly to a fixed point in $\mathrm{Fix}(\Gamma)$.

Let us mention for the non-experts that the weak convergence of a sequence in $\ell_2(\mathcal{I})$ means that all the components are converging, but not necessarily with the same speed. Strong convergence instead essentially means that all the components converge simultanously.

Eventually we have the weak convergence of the algorithm.

Theorem 14 (Weak convergence). *For any initial choice $x^{(0)} \in \ell_2(\mathcal{I})$, Algorithm ISTA produces a sequence $(x^{(n)})_{n\in\mathbb{N}}$ which converges weakly to a minimizer of $\mathcal{J}$.*

Proof. It is sufficient to observe that, due to our previous results, Lemma 11, Lemma 10, and Proposition 1, and the assumption $\|\Phi\| < 1$, the map

$$\Gamma(x) = \mathbb{S}_\lambda\left(x + \Phi^* y - \Phi^*\Phi x\right)$$

fulfills the requirements of Opial's Theorem. □

In the finite dimensional case, weak convergence implies also strong convergence of the algorithm. However, to prove also strong convergence in the infinite dimensional setting, one has to combine [21, Lemma 3.15, 3.17, and 3.18] with the weak convergence result of Theorem 14.

Corollary 2 (Strong convergence). *For any initial choice $x^{(0)} \in \ell_2(\mathcal{I})$, Algorithm ISTA produces a sequence $(x^{(n)})_{n\in\mathbb{N}}$ which converges strongly to a minimizer x^* of $\mathcal{J}$.*

3.1.4 Related algorithms

There exist by now several iterative methods that can be used for the minimization problem (114) in *finite dimensions*. We shall account a few of the most recently analyzed and discussed:

(a) the *Decreasing Iterative Soft Thresholding Algorithm* (D-ISTA) works like ISTA but uses a sequence of decreasing positive thresholds $\lambda^{(n+1)} \leq \lambda^n$, [19].
(b) the *GPSR-algorithm* (gradient projection for sparse reconstruction), another iterative projection method, in the auxiliary variables $v, z \geq 0$ with $x = v - z$, [29].
(c) the $\ell_1 - \ell_s$ *algorithm*, an interior point method using preconditioned conjugate gradient substeps (this method solves a linear system in each outer iteration step), [44].

(d) *FISTA* (fast iterative soft-thresholding algorithm) is a variation of the iterative soft-thresholding [5]. Define the operator $\Gamma(x) = \mathbb{S}_\lambda(x + \Phi^*(y - \Phi x))$. The FISTA is defined as the iteration, starting for $u^{(0)} = 0$,

$$x^{(n+1)} = \Gamma\left(x^{(n)} + \frac{t^{(n)} - 1}{t^{(n+1)}}\left(x^{(n)} - x^{(n-1)}\right)\right),$$

where $t^{(n+1)} = \frac{1+\sqrt{1+4(t^{(n)})^2}}{2}$ and $t^{(0)} = 1$.

As is addressed in the recent paper [49] which accounts a very detailed comparison of these different algorithms, they do perform quite well when the regularization parameter λ is sufficiently large, with a small advantage for GPSR. When λ gets quite small all the algorithms, except for FISTA, deteriorate significantly their performances. Moreover, local conditioning properties of the linear operator Φ seem particularly affecting the performances of iterative algorithms. Our numerical tests in Section 4.3, we observe that in a finite dimensional setting even FISTA is slow for small values of λ, compared to IHT or modified IRLS methods.

While these other methods are particularly suited for finite dimensional problems, it would be interesting to produce an effective strategy, for any range of the parameter λ, for a large class of infinite dimensional problems. In the paper [19] the following ingredients are combined for this scope:

- *multiscale preconditioning* allows for local well-conditioning of the matrix Φ and therefore reproduces at infinite dimension the conditions of best performances for iterative algorithms;
- *adaptivity* combined with a *decreasing thresholding strategy* allow for a *controlled* inflation of the support size of the iterations, promoting the minimal computational cost in terms of number of algebraic equations, as well as from the very beginning of the iteration the exploitation of the local well-conditioning of the matrix Φ.

In [63] the authors propose also an adaptive method similar to [19] where, instead of the soft-thresholding, a *coarsening function*, i.e., a compressed hard-thresholding procedure, is implemented. The emphasis in the latter contribution is on the regularization properties of such an adaptive method which does not dispose of a reference energy functional (112).

4 Numerical Results

In this section, we illustrate the theoretical results of Sections 2 and 3 by several numerical experiments and compare all presented methods. In Section 4.2 we focus on compressed sensing problems and thus compare the methods IRLS, CG-IRLS, and IHT-k. In Section 4.3, we compare IRLS-λ, CG-IRLS-λ and FISTA which are methods to solve the regularized problem. The comparison of IRLS with the first-order methods IHT-k and FISTA is fair since they do also exploit fast matrix-vector multiplications. We observe that it depends on the scope of the application, which method should be preferred. For instance, in some cases a modified IRLS is able to outperform IHT and FISTA, especially in high dimensional problems ($N \geq 10^5$). These results are somehow both surprising and counterintuitive as it is well-known that first order methods should be preferred in higher dimension. However, they can be easily explained by observing that in certain regimes preconditioning in the conjugate gradient method (as we show at the end of Subsection 4.3) turns out to be extremely efficient. Such benefits of preconditioning in IRLS have been reported already in minimization problems involving total variation terms [74]. Another significant outcome of our experiments is

that CG-IRLS is more robust than IHT-k in terms of higher successful recovery rate of less sparse solutions. This will be demonstrated by showing the corresponding phase transition diagrams of empirical success rates (Figure 6).

Before going into the detailed presentation of the numerical tests, we raise two plain numerical disclaimers concerning the numerical stability of CG-IRLS and CG-IRLS-λ:

- The first issue concerns IRLS methods in general: The case where $\varepsilon^n \to 0$, and $x_i^n \to 0$, for some $i \in \{1,\dots,N\}$ and $n \to \infty$, is very likely since our goal is the computation of sparse vectors. In this case w^n will at some n become too large to be properly represented by a computer. Thus, in practice, we have to provide a lower bound for ε by some $\varepsilon^{\min} > 0$. Imposing such a limit has the theoretical disadvantage that in general the algorithms are only calculating an approximation of the respective problems (4) and (26). Therefore, to obtain a "sufficiently good" approximation, one has to choose $\varepsilon^{\min}$ sufficiently small. This raises yet another numerical issue: If we choose, e.g., $\varepsilon^{\min} = 1\text{E-}8$ and assume that also $x_i^n \ll 1$, then w^n is of the order 1E+8. Compared to the entries of the matrix Φ, which are of the order 1, any multiplication or addition by such a value will cause serious numerical errors. In this context we cannot expect that the IRLS method reaches high accuracy, and saturation effects of the error are likely to occur before machine precision.
- The second issue concerns in particular the CG method: In Algorithm 8 and Algorithm 9 we have to divide at some point by $\left\|T^* p^i\right\|_{\ell_2}^2$ or $\langle A p^i, p^i\rangle_{\ell_2}$ respectively. As soon as the residual decreases, also p^i decreases with the same order of magnitude. If the above vector products are at the level of the machine precision, e.g. 1E-16, this would mean that the norm of the residual is of the order of its square-root, here 1E-8. But this is the measure of the stopping criterion. Thus, if we ask for a higher precision of the CG method, the algorithm might become numerically unstable.

In the following, we start with a description of the general test settings, which will be common for both Section 4.2 and 4.3.

4.1 Test Settings

All tests are performed with MATLAB version R2014a. To exploit the advantage of fast matrix-vector multiplications and to allow high dimensional tests, we use randomly sampled partial discrete cosine transformation matrices Φ. We perform tests in three different dimensional settings (later we will extend them to higher dimension) and choose different values N of the dimension of the signal, the amount m of measurements, the respective sparsity k of the synthesized solutions, and the index K in Algorithm (CG-)IRLS:

	Setting A	**Setting B**	**Setting C**
N	2000	4000	8000
m	800	1600	3200
k	30	60	120
K	50	100	200

For each of these settings, we draw at random a set of 100 synthetic problems on which a speed-test is performed. One synthetic problem is generated as follows. We first pick a permutation of the numbers $1,\dots,N$ at random. The first k entries of this permutation determine the support Λ. Then we draw the sparse

vector x^* at random with entries $x_i^* \sim \mathcal{N}(0,1)$ for $i \in \Lambda$ and $x^*_{\Lambda^c} = 0$, and a randomly row sampled non-normalized discrete cosine matrix Φ, where the full non-normalized discrete cosine matrix is given by

$$\Phi^{\text{full}}_{i,j} = \begin{cases} 1, & i = 1, j = 1, \dots, N, \\ \sqrt{2}\cos\left(\frac{\pi(2j-1)(i-1)}{2N}\right), & 2 \le i \le N, 1 \le j \le N. \end{cases}$$

For a given noise vector e of entries $e_i \sim \mathcal{N}(0,\sigma^2)$, we eventually obtain the measurements $y = \Phi x^* + e$. Later we need to specify the noise level and we will do so by fixing a signal to noise ratio. By assuming that Φ has the RIP of order k, we can estimate the measurement signal to noise ratio by

$$\text{MSNR} := \frac{\mathbb{E}(\|\Phi x^*\|_{\ell_2})}{\mathbb{E}(\|e\|_{\ell_2})} \sim \frac{\sqrt{k}}{\sqrt{m}\sigma}.$$

In practice, we set the MSNR first and choose the noise level $\sigma = \frac{\sqrt{k}}{\text{MSNR}\sqrt{m}}$. If $\text{MSNR} = \infty$, the problem is noiseless, i.e., $e = 0$.

4.2 Comparison of Numerical Methods for Compressed Sensing

4.2.1 Specific Settings

We restrict the maximal number of outer IRLS iterations to 15. Furthermore, we modify (72), so that the CG-algorithm also stops as soon as $\left\|\rho^{n+1,i}\right\|_{\ell_2} \le 1\text{E-}12$. As soon as the residual undergoes this particular threshold, we call the CG solution (numerically) "exact". The ε-update rule is extended by imposing the lower bound $\varepsilon^n = \varepsilon^n \vee \varepsilon^{\min}$ where $\varepsilon^{\min} = 1\text{E-}9/N$. The summable sequence $(a_n)_{n\in\mathbb{N}}$ in Theorem 8 is defined by $a_n = 100 \cdot (1/2)^n$.

As we define the synthetic tests by choosing the solution x^* of the linear system $\Phi x^* = y$ (here we assume $e = 0$), we can use it to determine the error of the iterations $\|\tilde{x}^n - x^*\|_{\ell_2}$.

4.2.2 A First Experiment

To get an immediate impression about the general behavior of CG-IRLS, we compare its performance in terms of accuracy and speed to IRLS, where the intermediate linear systems are solved exactly by the standard MATLAB backslash operator.

In this first single trial experiment, we choose an instance of setting B, and set $\tau = 1$ for CG-IRLS and compare it to IRLS with different values of τ. The result is presented in the left plot of Figure 4. We show the decrease of the relative error in ℓ_2-norm as a function of the computational time. One sees that the computational time of IRLS is significantly outperformed by CG-IRLS and by the exploitation of fast matrix-vector multiplications. The standard IRLS is not competitive in terms of computational time, even if we choose $\tau < 1$, which is known to yield super-linear convergence [22]. With increasing dimension of the problem, in general the advantage of using the CG method becomes even more significant. Furthermore, such a straight-

forward implementation of Algorithm CG-IRLS does not outperform IHT-k in terms of computational time. We also observe the expected numerical error saturation (as mentioned at the beginning of this section), which appears as soon as the accuracy falls below 1E-13.

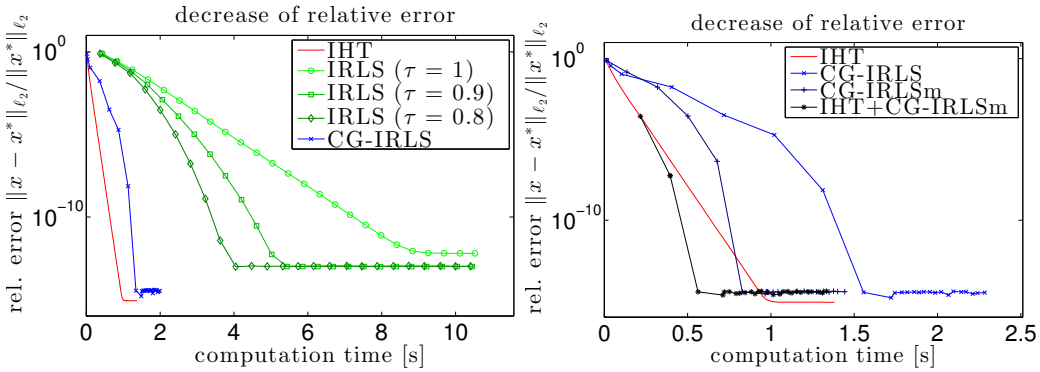


Fig. 4 Single trial of Setting B. Left: Relative error plotted against the computational time for IRLS[$\tau=1$] (light green, $\circ$), IRLS[$\tau=0.9$] (green, $\square$), IRLS[$\tau=0.8$] (dark green, $\Diamond$), CG-IRLS (blue, $\times$), and IHT-k (red, $-$). Right: Relative error plotted against computational time for CG-IRLS (blue, $\times$), CG-IRLSm (dark blue, $+$), IHT+CG-IRLSm (black, $*$), and IHT-k (red, $-$).

For this test, we set the parameter β in the ε-update rule to 2. We comment on the choice of this particular parameter in a dedicated paragraph below.

4.2.3 Modifications to CG-IRLS

As we have shown by means of a single trial in the previous paragraph, CG-IRLS as it is presented in Section 2.2.1.5 is not able to outperform IHT-k. Therefore, we introduce the following practical modifications to the algorithm:

1. We introduce the parameter `maxiter_cg`, which defines the maximal number of inner CG iterations. Thus, the inner loop of the algorithm stops as soon as `maxiter_cg` iterations were performed, even if the theoretical tolerance tol_n is not reached yet.
2. CG-IRLS includes a stopping criterion depending on tol_{n+1}, which is only *implicitly* given as a function of ε^{n+1} (compare Section 2.2.1.5, and in particular formulas (72) and (73)), which in turn depends on the current $\tilde{x}^{n+1}$ by means of sorting and a matrix-vector multiplication. To further reduce the computational cost of each iteration, we avoid the aforementioned operations by making tol_{n+1} *explicitly* dependent exclusively on ε^n.
3. The left plot of Figure 4 reveals that in the beginning CG-IRLS reduces the error more slowly than IHT-k, and it gets faster after it reached a certain ball around the solution. Therefore, we use IHT as a warm up for CG-IRLS, in the sense that we apply a number `start_iht` of IHT iterations to compute a proper starting vector for CG-IRLS.

We call *CG-IRLSm* the algorithm with modifications (i) and (ii), and *IHT+CG-IRLSm* the algorithm with modifications (i), (ii), and (iii). We set `maxiter_cg` $=\lfloor m/12 \rfloor$, `start_iht` $=150$, and we set β to 0.5.

If these algorithms are executed on the same trial as in the previous paragraph, we obtain the result which is shown on the right plot in Figure 4. For this trial, the modified algorithms show a significantly reduced computational time with respect to the unmodified version and they converge now faster than IHT-k. However, the introduction of the practical modifications (i)–(iii) does not necessarily comply anymore with the assumptions of Theorem 8. This means that, additionally to the possible violation of the Null Space Property, we have to take into consideration that the modified methods can fail more often in the recovery of a sparse vector, because of violation of the other aforementioned assumptions (72) and (73). In the next paragraph, we will empirically investigate the failure rate, and explore the performance of the different methods on a sufficiently large test set.

Another natural modification to CG-IRLS is the introduction of a preconditioner to compensate for the deterioriation of the condition number of $\Phi D_n \Phi^*$ as soon as ε^n gets too small (in turn as w^n gets too large). The matrix $\Phi\Phi^*$ is very well conditioned, while the matrix $\Phi D_n \Phi^*$ "sandwiching" D_n gets more and more ill-conditioned for n larger and larger, and, unfortunately, it is hard to identify additional "sandwiching" preconditioners P_n yielding the matrix $P_n \Phi D_n \Phi^* P_n^*$, really able to compensate for the spoiled spectrum. In the numerical experiments standard preconditioners failed to yield any significant improvement in terms of convergence speed. Hence, we do not insist in further introducing a preconditioner in this case.
Instead, as we will show at the end of Subsection 4.3, a standard preconditioning (actually we use just a plain Jacobi preconditioning) of the matrix

$$\left(\Phi^*\Phi + \operatorname{diag}\left[\lambda\tau w_j^n\right]_{j=1}^N\right),$$

where the source of singularity is added to the product $\Phi^*\Phi$, leads to a dramatic improvement of computational speed.

4.2.4 Empirical Test on Computational Time and Failure Rate

In the following, we define a method to be "successful" if it is computing a solution x for which the relative error $\|x - x^*\|_{\ell_2} / \|x^*\|_{\ell_2} \leq 1\text{E-}13$. The computational time of a method is measured by the time it needs to produce the first iterate which reaches this accuracy. In the following, we present the results of a test which runs the methods CG-IRLS, CG-IRLSm, IHT+CG-IRLSm, and IHT-k on 100 trials of Setting A, B, and C respectively and $\tau \in \{1, 0.9, 0.8\}$. For values of $\tau < 0.8$ the methods become unstable, due to the severe nonconvexity of the problem and we assume that no good performances can be reached under this level. Therefore we do not investigate further these cases. Let us stress that IHT-k does not depend on τ.

In each setting we check for each trial which methods succeeded or failed. If all methods succeeded, we compared the computational time, determined the fastest method, and counted the computational time of each method for the respective mean computational time. The results are shown in Figure 5. By analyzing the diagrams, we are able to distill the following observations:

- In particular in Setting A and B, CG-IRLSm and IHT+CG-IRLSm are better or comparable to IHT-k in terms of the mean computational time and provide in most cases the fastest method. CG-IRLS performs much worse. The failure rate of all the methods is here negligible.
- The gap in the computational time between all methods becomes larger when N is larger.
- With increasing dimension of the problem, the advantage of using the modified CG-IRLS methods subsides, in particular in Setting C.

- In the literature [13–15, 22] superlinear convergence is actually reported for $\tau < 1$, and perhaps one of the most surprising outcomes is that the best results for all CG-IRLS methods are instead obtained for $\tau = 1$. This is easily explained by observing that superlinear convergence kicks in only in a rather small ball around the solution and hence does not necessarily improve the actual computational time!
- Not only the computational advantage of superlinear convergence is not realized for $\tau < 1$, but also the failure rate of the CG-IRLS based methods increases with decreasing τ. However, as expected, CG-IRLS does not fail in the convex case of $\tau = 1$. The failure of CG-IRLS for $\tau < 1$ can be attributed to nonconvexity. The failure rate of the modified methods is even higher due to the possible violation of the assumptions in Theorem 8.

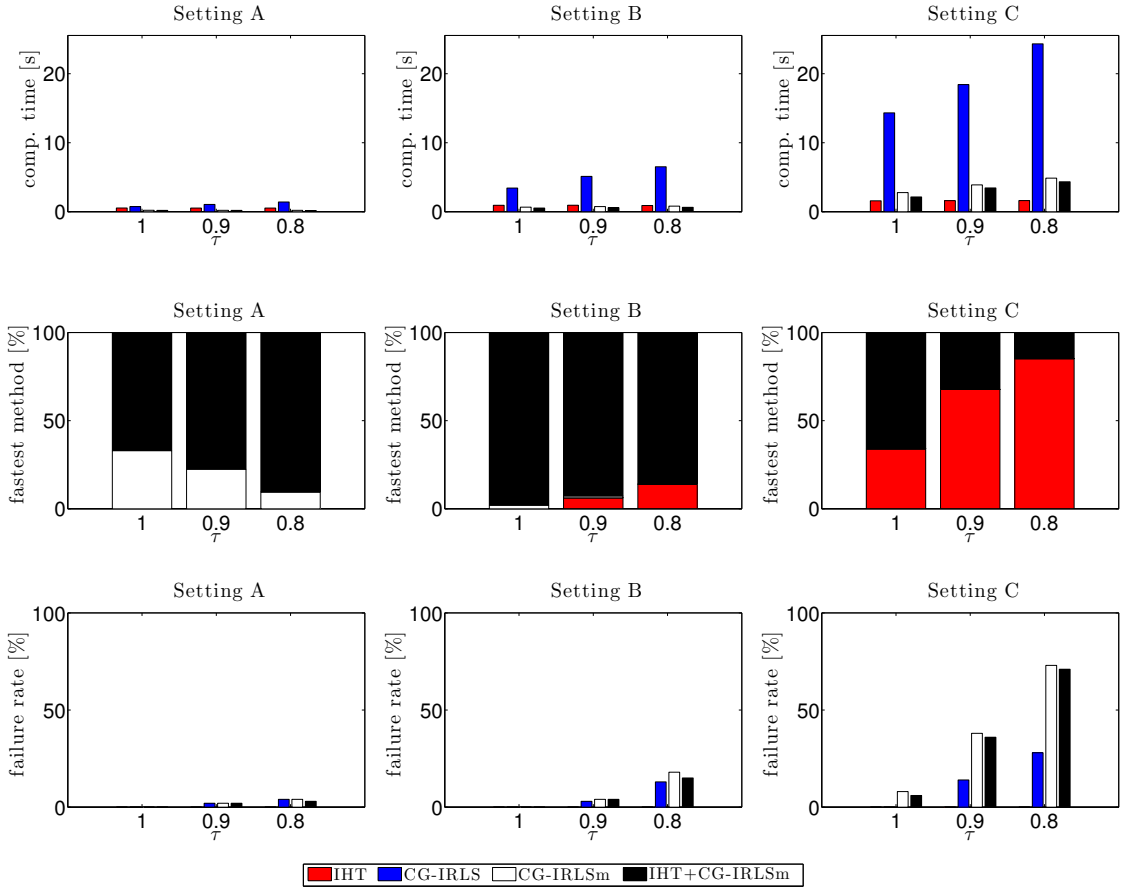


Fig. 5 Empirical test on Setting A, B, and C for the methods CG-IRLS (blue), CG-IRLSm (white), IHT+CG-IRLSm (black), and IHT-k (red). Upper: Mean computational time. Center: Fastest method (in %). Lower: Failure rate (in %).

We conclude that CG-IRLSm and IHT+CG-IRLSm perform well for $\tau = 1$ and for the problem dimension N within the range of 1000 – 10000. They are even able to outperform IHT-k. However, by extrapolation of

the numerical results IHT-k is expected to be faster for $N > 10000$. (This is actually in compliance with the general folklore that first order methods should be preferred for higher dimension.) As soon as $N < 1000$, direct methods as Gaussian elimination are faster than CG, and thus, one should use standard IRLS with $\tau < 1$ or IHT-k.

4.2.5 Choice of β, `maxiter_cg`, and `start_iht`

The numerical tests in the previous paragraph were preceded by a careful and systematic investigation of the tuning of the parameters β, `maxiter_cg`, and `start_iht`. While we fixed `start_iht` to 100, 150, and 200 for Setting A, B, and C respectively to produce a good starting value, we tried $\beta \in \{1/N, 0.01, 0.1, 0.5, 0.75, 1, 1.5, 2, 5, 10\}$, and `maxiter_cg` $\in \{\lfloor m/8 \rfloor, \lfloor m/12 \rfloor, \lfloor m/16 \rfloor\}$ for each setting. The results of this parameter sensitivity study can be summarized as follows:

- The best computational time is obtained for $\beta \sim 1$. In particular the computational time is not depending substantially on β in this order of magnitude. More precisely, for CG-IRLS the choice of $\beta = 0.5$ and for (IHT+)CG-IRLSm the choice of $\beta = 2$ works best.
- The choice of `maxiter_cg` very much determines the tradeoff between failure and speed of the method. The value $\lfloor m/12 \rfloor$ seems to be the best compromise. For a smaller value the failure rate becomes too high, for a larger value the method is too slow.

4.2.6 Phase Transition Diagrams

Beside the empirical analysis of the speed of convergence, we also want to investigate the robustness of CG-IRLS with respect to the achievable minimal sparsity level for exact recovery of x^*. Therefore, we fix $N = 2000$ and we compute a phase transition diagram for IHT-k and CG-IRLS on a regular Cartesian 50×40 grid, where one axis represents m/N and the other represents k/m. For each grid point we plot the empirical success recovery rate, which is numerically realized by running both algorithms on 20 random trials. CG-IRLS or IHT-k is successful if it is able to compute a solution with a relative error of less than 1E-4 within 20 or 500 (outer) iterations respectively. Since we want to simulate a setting in which the sparsity k is not known exactly, we set the parameter $K = 1.1 \cdot k$ for both IHT-k and CG-IRLS. The interpolated plot is shown in Figure 6. It turns out that CG-IRLS has a significantly higher success recovery rate than IHT-k for less sparse solutions.

4.3 Comparison of Numerical Methods for Sparse Recovery

4.3.1 Specific Settings

We restrict the maximal number of outer IRLS iterations to 25. Furthermore, we modify (95), so that the CG-algorithm also stops as soon as $\left\|\rho^{n+1,i}\right\|_{\ell_2} \leq 1\text{E-}16 \cdot N^{3/2} m$. As soon as the residual undergoes this particular threshold, we call the CG solution (numerically) "exact". The ε-update rule is extended by imposing the lower bound $\varepsilon^n = \varepsilon^n \vee \varepsilon^{\min}$ where $\varepsilon^{\min} = 1\text{E-}9$. Additionally we propose to choose $\varepsilon^{n+1} \leq 0.8^n \varepsilon^n$, which practically turns out to increase dramatically the speed of convergence. The summable sequence $(a_n)_{n \in \mathbb{N}}$ in

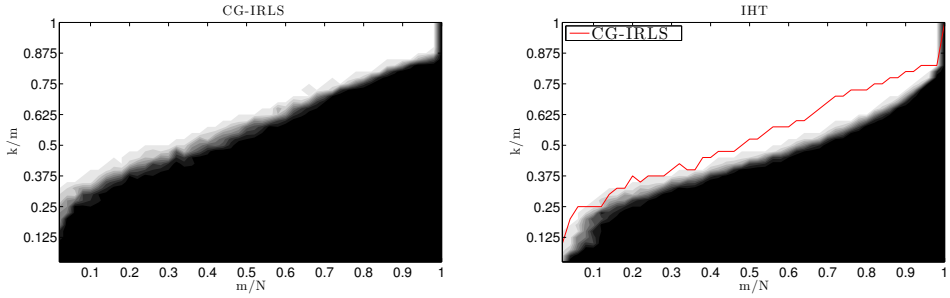


Fig. 6 Phase transition diagrams of IHT-k and CG-IRLS for $N = 2000$. The recovery rate is presented in grayscale values from 0% (white) up to 100% (black). As a reference, in the right subfigure, the 90% recovery rate level line of the CG-IRLS phase transition diagram is plotted to show more evidently the improved success rate of the latter algorithm.

Theorem 10 is defined by setting $a_n = \sqrt{Nm} \cdot 1\text{E}{+}4 \cdot (1/2)^n$. We split our investigation into a noisy and a noiseless setting.

For the noisy setting we set MSNR $= 10$. According to [9], we choose $\lambda = c\sigma\sqrt{m \log N}$ as a near-optimal regularization parameter, where we empirically determine $c = 0.48$. Since we work with relatively large values of λ in the regularized problem (26), we cannot use the synthesized sparse solution x^* as a reference for the convergence analysis. Instead, we use the reliable method FISTA to compute the minimizer of the convex functional (26). In the nonconvex case of $\tau < 1$, there is no method which guarantees the computation of the global minimizer of (89), thus, we have to omit a detailed speed-test in this case. However, we describe the behavior of Algorithm CG-IRLS-λ for τ changing.

If the problem is noiseless, i.e., $e = 0$, the solution x^λ of (26) converges to the solution of (4) for $\lambda \to 0$. Thus, ins this case we choose $\lambda = m \cdot 1\text{E}{-}8$, and assume the synthesized sparse solution x^* as a good proxy for the minimizer and a reference for the convergence analysis. (Actually, this can also be seen the other way around, i.e., we use the minimizer x^λ of the regularized functional to compute a good approximation to x^*.) In fact, for $\lambda \approx 0$, as we comment below in more detail, FISTA is basically of no use.

4.3.2 A First Experiment

As in the previous subsection, we want to show first that the CG-method within IRLS-λ produces significant improvements in terms of the computational speed. Therefore we choose a noisy trial of Setting B, and compare the computational time of the methods IRLS-λ, CG-IRLS-λ, and FISTA. The result is presented on the left plot of Figure 7. We observe, that CG-IRLS-λ computes the first iterations in much less time than IRLS-λ, but due to bad conditioning of the inner CG problems it performs much worse afterwards. Furthermore, we confirm the expectation that the algorithm is not suitable to compute a highly accurate solution. For the computation of a solution with a relative error in the order of 1E-3, CG-IRLS-λ outperforms FISTA. Indeed, FISTA is able to compute a highly accurate solution, nevertheless a solution with a relative error of 1E-3 should be sufficient in most applications since the goal in general is not to compute the minimizer of the Lagrangian functional but an approximation of the sparse signal.

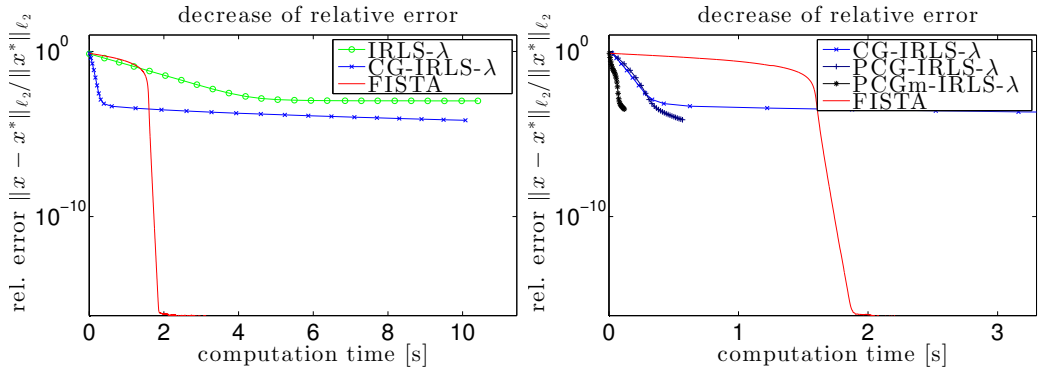


Fig. 7 Single trial of Setting B. Left: Relative error plotted against the computational time for IRLS-λ (light green, $\circ$), CG-IRLS-λ (blue, $\times$), and FISTA (red, $-$). Right: Relative error plotted against computational time for CG-IRLS-λ (blue, $\times$), PCG-IRLS-λ (dark blue, $+$), PCGm-IRLS-λ (black, $*$), and FISTA (red, $-$).

4.3.3 Modifications to CG-IRLS-λ

To further decrease the computational time of CG-IRLS-λ, we propose the following modifications:

1. To overcome the bad conditioning in the CG loop, we precondition the matrix $A_n = \Phi^*\Phi + \operatorname{diag}\left[\lambda\tau w_j^n\right]_{j=1}^N$ by means of the Jacobi preconditioner, i.e., we pre-multiply the linear system by the inverse of its diagonal, $(\operatorname{diag} A_n)^{-1}$, which is a very efficient operation in practice.
2. We introduce the parameter `maxiter_cg` which defines the maximal number of inner CG iterations.

We call *PCG-IRLS-λ* the algorithm with modification (i), and *PCGm-IRLS-λ* the algorithm with modifications (i) and (ii). We set `maxiter_cg` $= 4$, and run these algorithms on the same trial of Setting B as in the previous paragraph. The respective result is shown on the right plot of Figure 7. This time, preconditioning effectively yields a strong decrease of computational time, especially in the final iterations where A_n is badly conditioned. Furthermore, modification (ii) importantly increases the performance of the proposed algorithm also in the initial iterations. However, again we have to take into consideration that we may violate the assumptions of Theorem 10 by using (ii), and therefore the method PCGm-IRLS-λ is expected to fail in some cases. In the following two paragraphs, we present simulations on noisy and noiseless data, which give a more precise picture of the speed and failure rate of the previously introduced methods in comparison to FISTA and IHT-k.

4.3.4 Empirical Test on Computational Time and Failure Rate with Noisy Data

In the previous paragraph, we observed that the CG-IRLS-λ methods are only computing efficiently solutions with a low relative error. Thus we now focus on this setting and compare the three methods PCG-IRLS-λ, PCGm-IRLS-λ, and FISTA with respect to their computational time and failure rate in recovering solutions with a relative error of 1E-1, 1E-2, and 1E-3. We only consider the convex case $\tau = 1$. Similarly to the procedure in Section 4.2, we run these algorithms on 100 trials for each setting with the respectively chosen values of λ. In Figure 8 the upper bar plot shows the result for the mean computational time and the

lower stacked bar plot shows how often a method was the fastest one. We do not present a plot of the failure rate since none of the methods failed at all. By means of the plots, we demonstrate that both PCG-IRLS-λ, and PCGm-IRLS-λ are faster than FISTA, while PCGm-IRLS-λ always performs best.

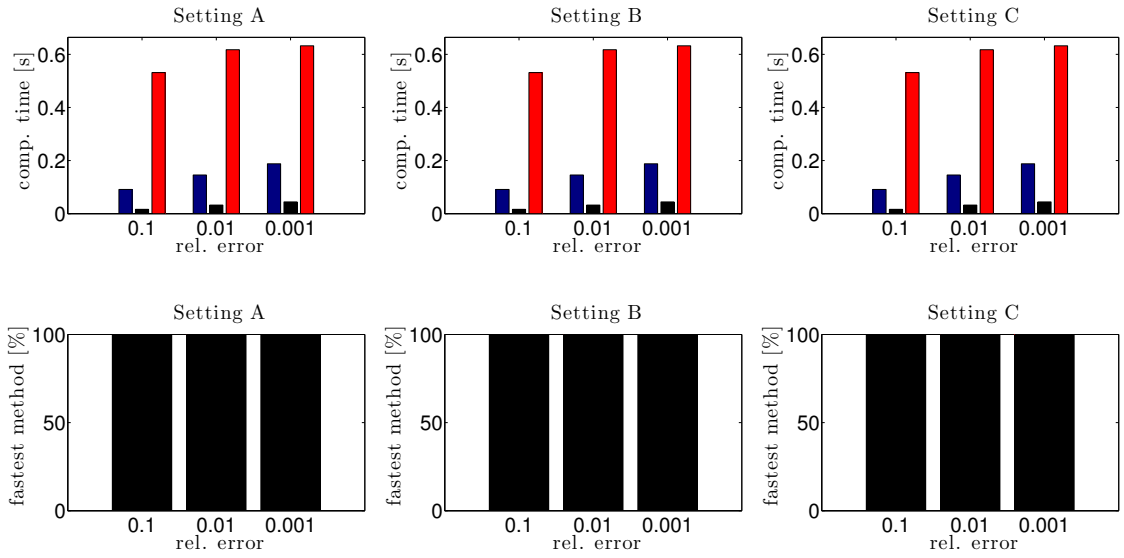


Fig. 8 Empirical test on Setting A, B, and C for the methods PCG-IRLS-λ (blue), PCGm-IRLS-λ (black), and FISTA (red). Upper: Mean computational time. Lower: Fastest method (in %).

4.3.5 Empirical Test on Computational Time and Failure Rate with Noiseless Data

In the noiseless case, we compare the computational time of FISTA and PCGm-IRLS-λ to IHT-k and IHT+CG-IRLSm. We set `maxiter_cg` $= 40$ for PCGm-IRLS-λ. In a first test, we run these algorithms on one trial of Setting A, and C respectively, and plot the results in Figure 9.

As already mentioned, FISTA is not suitable for such a small value of λ and converges extremely slowly, but PCGm-IRLS-λ can compete with the remaining methods. While IHT+CG-IRLSm is not able to outperform IHT-k, PCGm-IRLS-λ is not only as fast as IHT-k but also seems to become faster than IHT-k with increasing dimension. Because of this observation we formulate the conjecture that PCGm-IRLS-λ will also provide the fastest results also in rather high dimensional problems. To validate this hypothesis numerically, we introduce two new high dimensional settings:

	Setting D	**Setting E**
N	100000	1000000
m	40000	400000
k	1500	15000
K	2500	25000

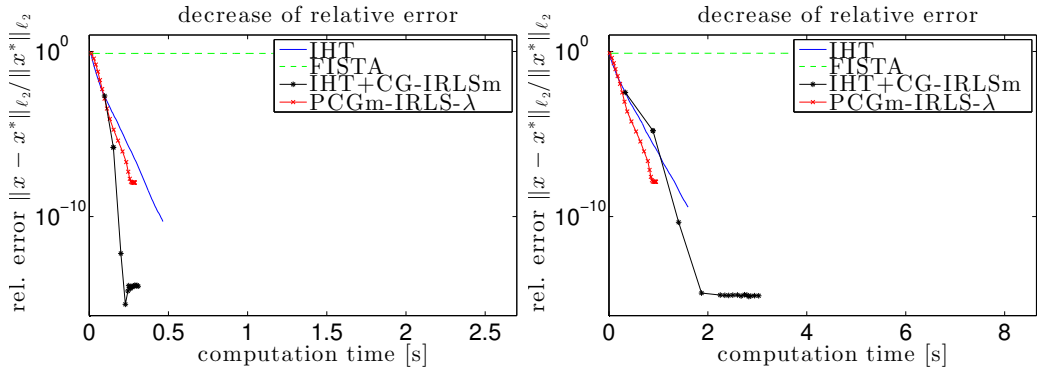


Fig. 9 Left: Setting A. Right: Setting C. Comparison of IHT-k (blue, $-$), FISTA (green, $--$), IHT+CG-IRLSm (black, $*$), and PCGm-IRLS-λ (red, $\times$).

We run IHT-k and PCGm-IRLS-λ, which are the most promising algorithms, on a trial of the large scale settings D and E. The result, which is plotted in Figure 10, shows that PCGm-IRLS-λ is able to outperform IHT-k on this large scale as long as one does not wish for an extremely low relative error ($\leq$ 1E-8), because of the error saturation effect. We confirm this outcome in a test on 100 trials for Setting D and E and present the result in Figure 11.

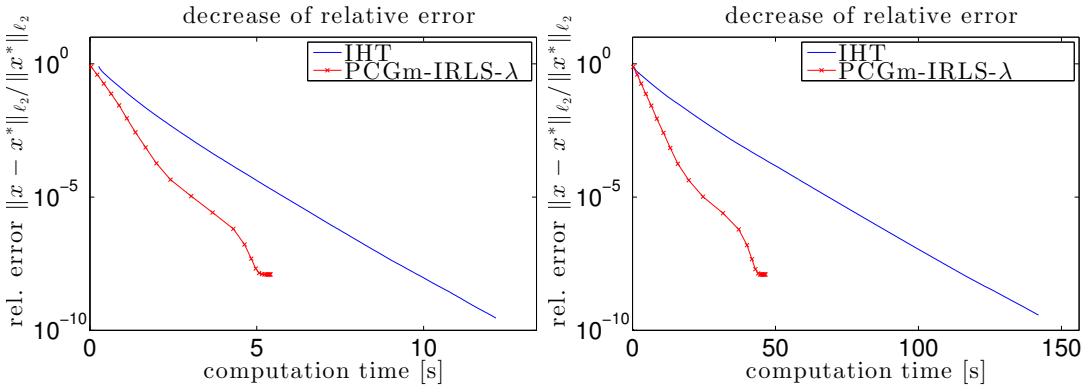


Fig. 10 Left: Setting D. Right: Setting E. Comparison of IHT-k (blue, $-$), and PCGm-IRLS-λ (red, $\times$).

4.3.6 Dependence on τ

In the last experiment of this paper, we are interested in the influence of the parameter τ. Of course, changing τ also means modifying the problem and setting a different minimizer as well as the appearing of possible local minimizers. Therefore we do not compare the speed of the method to FISTA. In Figure 12,

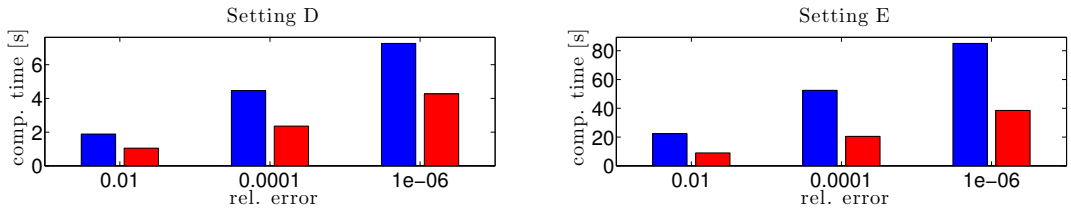


Fig. 11 Empirical test on the mean computational time of Setting D and E for the methods IHT-k (blue), and PCGm-IRLS-λ (red).

we show the performance of Algorithm PCGm-IRLS-λ for a single trial of Setting C and the parameters $\tau \in \{1, 0.9, 0.8, 0.7\}$ for the noisy and noiseless setting. As reference for the error analysis, we chose the sparse synthetic solution x^*, which is actually not the solution of the problem.

In both the noisy and noiseless setting, using a parameter $\tau < 1$ improves the computational time of the algorithm. In the noiseless case $\tau = 0.9$ seems to be a good choice, smaller values do not improve the performance. Contrarily, in the noisy setting the computational time decreases with decreasing τ.

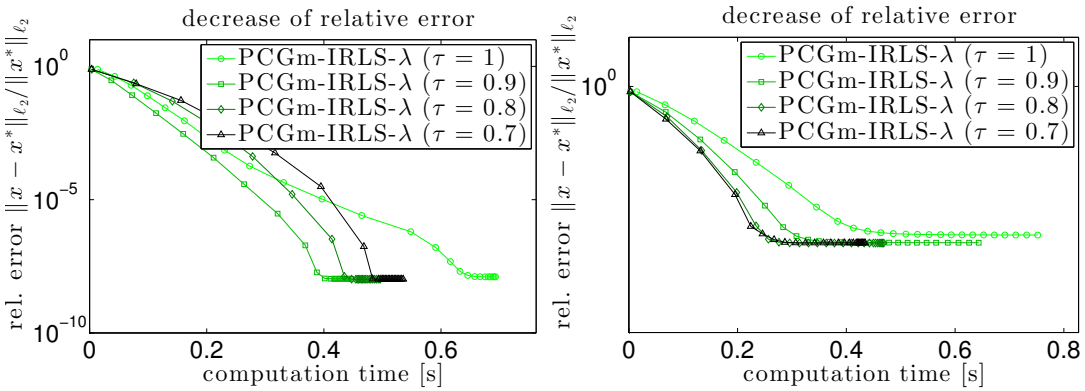


Fig. 12 Results of Algorithm PCGm-IRLS-λ for a single trial of Setting C for different values of τ with noise (right) and without noise (left).

5 Noise Folding in Compressed Sensing

So far, all methods that we considered were particularly suited the noiseless model (19), or the model (20), which takes into account the noise on the measurements. However, as already briefly explained in the introductory Section 1.4.2, additional noise on the signal, as it is considered by the model (30), will cause the noise-folding phenomenon and may severely influence the recovery results.

The noise folding phenomenon was first noted in compressed sensing by [12,70], which follow a different approach while getting to the same result. To simplify the presentation and focus on the relevant effects, we will from now on only consider only the signal noise $n \neq 0$ and set the measurement noise $e = 0$. In [12], the authors assume the signal noise to be white, i.e., $n \sim (\mathcal{N}(0,\sigma_n))^N$. This in general means that Φn is not white and has covariance $C_{\Phi n} = \sigma_n^2 \Phi\Phi^*$. Multiplying the linear system left-hand by the matrix $M := \left(\frac{m}{N} C_{\Phi n}\right)^{-1}$ transforms it into the system

$$\bar{y} = \bar{\Phi} x + \bar{e},$$

where $\bar{y} := My$, $\bar{\Phi} = M\Phi$, and $\bar{e} := M\Phi n$. We actually performed a whitening of Φn, so that now $\bar{e} \sim \left(\mathcal{N}(0, \sqrt{\frac{N}{m}}\sigma_n)\right)^m$. Concerning the properties of the matrix $\bar{\Phi}$, we cite

Lemma 12 ([12, Proposition 1]). *Assume that $\kappa := \left\|I - \frac{m}{N}\Phi\Phi^*\right\| < \frac{1}{2}$ and that Φ satisfies the RIP of order k with constant δ_k. Then $\hat{\Phi}$ satisfies the RIP of order k with constant $\bar{\delta}_k := \max\{1-(1-\delta_k)\sqrt{1-\kappa_1}, (1+\delta_k)\sqrt{1+\kappa_1}-1\}$, with $\kappa_1 := \kappa/(1-\kappa)$.*

Remark 5. Note that the assumption $\left\|I - \frac{m}{N}\Phi\Phi^*\right\| < \frac{1}{2}$ can be fulfilled with high probability in the standard setting of compressed sensing, e.g., if the entries of Φ are i.i.d. Gaussian (compare [73, Corollary 35, Theorem 39]).

Thus, we conclude that a linear measurement process which is corrupted by white noise on the signal with entries of standard deviation σ_n is equivalent to a linear measurement process whose RIP is close to the one of the original process, where there is no signal noise present but noise on the measurement with entries of standard deviation $\sqrt{\frac{N}{m}}\sigma_n$.

In [70] the authors approach the problem by the following idea: Assume that there is an oracle that provides us with the support of the sparse signal $\Lambda = \operatorname{supp}(\bar{x})$. Then a natural recovery strategy is

$$\underset{\operatorname{supp}(z)=\Lambda}{\arg\min} \|\Phi z - y\|_{\ell_2}. \tag{126}$$

Theorem 15 ([70, Theorem 4.3]). *Let x^* be the solution to problem* (126) *(assume Φ to have full rank). Suppose n to be white noise, and Φ satisfies the RIP of order k with constant δ_k and to have orthogonal rows, each of norm $\sqrt{\frac{N}{m}}$. Then an error estimate for x^* is given by*

$$\frac{N}{m}(1+\delta_k)^{-1}\mathbb{E}(\|n_\Lambda\|_{\ell_2}^2) \leq \frac{N}{m}(1+\delta_k)^{-1}\mathbb{E}(\|\bar{x}-x^*\|_{\ell_2}^2) \leq \frac{N}{m}(1-\delta_k)^{-1}\mathbb{E}(\|n_\Lambda\|_{\ell_2}^2).$$

Remark 6. The condition on Φ that it consists of orthogonal rows of equal norm is not restrictive in the setting of compressed sensing since for any arbitrary matrix $\tilde{\Phi}$ which satisfies the RIP, it is always possible to construct a matrix Φ that has the same row space as $\tilde{\Phi}$ and does satisfy these properties (compare [70, Lemma 4.1]).

Also this result shows the nature of the noise-folding phenomenon. The ℓ_2-norm error of the recovered signal with respect to the original signal is in the order of $\sqrt{\frac{N}{m}}$ times the norm of the noise. Notice that there is no guarantee that (126) is the best decoder, once the support Λ is known, but up to this point the previous result shows that the actual challenge in the noise-folding regime is to identify a decoder which "simulates" the oracle, i.e., which robustly and reliably determines the support Λ.

Pursuant to the latter observation, we wish to focus in this section on two fundamental consequences of noise folding: the loss of accuracy in the recovery of the relevant entries of the original vector $\bar{x}$, and the correct detection of their index support.

We restrict ourselves to a purely deterministic setting. First of all, we show that, unfortunately, the classical ℓ_1-minimization, but also the iteratively re-weighted ℓ_1-minimization [11, 54], considered one of the most robust recovery methods, easily fails in both the tasks mentioned above. The deep reason of this failure is the lack of selectivity of these algorithms, which are designed to promote not only the sparsity of the signal x but also of the recovered noise. This has the consequence, as a sort of balancing principle, that the miscomputed components of the noise n, if it is not originally of impulsive nature, blow up the inaccuracy in the detection and approximation of $\bar{x}$.

To overcome these difficulties of these popular methods, we propose two decoding procedure: The first procedure combines ℓ_1- and regularized *selective least p-powers* minimization (see Definition 4 below), and the second combines again a warm-up step based on ℓ_1-minimization with a nonconvex optimization realized by IHT as presented in Section 2.3.2, and a final correction step realized by a convex optimization. Both procedures are based on a *selectivity principle* which allows to reduce the noise component affecting the signal and enhance the support identification.

The remainder of this section is organized as follows. In Section 5.1, we shall describe the limitations of ℓ_1-minimization when noise on the signal is present, and we mention that very similarly an analogue analysis can be performed for the iteratively re-weighted ℓ_1-minimization based on the results in [54]. Afterwards, as an alternative, we propose two methods, i.e., the linearly constrained minimization of the regularized selective p-potential functional, and a method based on iterative hard thresholding, and show that certain *sufficient conditions* for recovery *indicate* a potentially better performance than the one provided by ℓ_1-minimization and iteratively re-weighted ℓ_1-minimization. Within our investigations with respect to noise-folding, we measure the *performance* of a method by its ability of identifying and approximating the relevant entries of the original signal, where the *relevant entries* of a signal are the ones which exceed in absolute value a predefined threshold $r > 0$. In particular the methods we propose perform better than the classical ℓ_1-minimization based ones, especially when $r > 0$ is given, but the number k of relevant entries of the original signal is not known. Finally, in Section 5.4, we report the results of numerical experiments, which we made to illustrate and support our theoretical guarantees, and the comparisons between all the mentioned decoding methods.

5.1 Support Identification Stability Results for ℓ_1-Minimization and Re-weighted ℓ_1-Minimization

For later use, let us denote, for $1 \leq p \leq 2$ and q such that $\frac{1}{p} + \frac{1}{q} = 1$,

$$\kappa_p := \kappa_p(N,k) := \begin{cases} 1, & p = 1, \\ \sqrt[q]{N-k}, & 1 < p \leq 2. \end{cases} \tag{127}$$

The following simple proposition shows how one can estimate the support of the relevant entries of the original signal if we know the support of the ℓ_1-minimizer.

Let us stress clearly that the best k-term approximation $\bar{x} = x_{[k]}$ of a signal $x \in \mathbb{R}^N$ actually models the relevant entries of the signal and the residual $n = x - x_{[k]}$ the noise affecting $\bar{x}$, so that we can write $x = \bar{x} + n$. Notice that we do not specify whether the entries of $\bar{x}$ are themselves affected by noise. Indeed, as we can at

most approximate them anyway with an accuracy, which is never better than the noise level $\eta = \|n\|_{\ell_2}$, see, (138), it is clearly redundant to discuss their exactness or noiseless nature.

Theorem 16. *Let $x \in \mathbb{R}^N$ be a noisy signal with k relevant entries and the noise level $\eta \in \mathbb{R}$, $\eta \geq 0$, i.e., for $\Lambda = \mathrm{supp}(x_{[k]})$,*

$$\sum_{j \in \Lambda^c} |x_j|^p \leq \eta^p, \tag{128}$$

for a fixed $1 \leq p \leq 2$. Consider further an encoder $\Phi \in \mathbb{R}^{m \times N}$ which has the (k, γ_k)-NSP, with $\gamma_k < 1$, the respective measurement vector $y = \Phi x \in \mathbb{R}^m$, and the ℓ_1-minimizer

$$x^* := \operatorname*{arg\,min}_{z \in \mathcal{F}_\Phi(y)} \|z\|_{\ell_1}.$$

If the i-th component of the original signal x is such that

$$|x_i| > \frac{2(1+\gamma_k)}{1-\gamma_k} \kappa_p \, \eta, \tag{129}$$

then $i \in \mathrm{supp}(x^*)$.

Proof. Hölder's inequality applied on the instance optimality property (5), and the assumption (128) yield the estimate

$$\|x^* - x\|_{\ell_1} \leq \frac{2(1+\gamma_k)}{1-\gamma_k} \sigma_k(x)_{\ell_1} \leq \frac{2(1+\gamma_k)}{1-\gamma_k} \kappa_p \, \eta. \tag{130}$$

We now choose a component $i \in \{1, \ldots, N\}$ such that $|x_i| > \frac{2(1+\gamma_k)}{1-\gamma_k} \kappa_p \, \eta$, and assume $i \notin \mathrm{supp}(x^*)$. This leads to the contradiction:

$$|x_i| = |x_i - x_i^*| \leq \|x - x^*\|_{\ell_1} \leq \frac{2(1+\gamma_k)}{1-\gamma_k} \kappa_p \, \eta < |x_i|. \tag{131}$$

Hence, necessarily $i \in \mathrm{supp}(x^*)$. □

The noise level substantially influences the ability of support identification. Here, the noisy signal should have (as a sufficient condition) the k largest entries in absolute value above

$$r_1 := \frac{2(1+\gamma_k)}{1-\gamma_k} \kappa_p \, \eta,$$

in order to guarantee support identification.

A recent ansatz to enhance the reconstruction of sparse vectors is *iteratively re-weighted ℓ_1-minimization* [11, 54]. It iteratively computes the solution of

$$z^{n+1} = \operatorname*{arg\,min}_{z \in \mathcal{F}_\Phi(y)} \sum_{i=1}^N w_i^n |z_i|, \quad n = 0, 1, 2, \ldots$$

while updating the weights according to $w_i^n = (|z_i^n| + a)^{-1}$ for all $i = 1, \ldots, N$, for a suitably chosen stability parameter $a > 0$ (which we consider fixed in the rest of this section). We denote the limit of this iterative decoding procedure by $\Delta_{1\mathrm{rew}}(y)$. We are able to show a similar support identification result also in the case of the iteratively re-weighted ℓ_1-minimization, as a consequence of the respective instance optimality result in [54, Theorem 3.2].

Theorem 17. *Let $x \in \mathbb{R}^N$ be a noisy signal with k relevant entries and the noise level $\eta \in \mathbb{R}$, $\eta \geq 0$, i.e., for $\Lambda = \mathrm{supp}(x_{[k]})$,*

$$\sum_{j \in \Lambda^c} |x_j|^p \leq \eta^p, \tag{132}$$

for a fixed $1 \leq p \leq 2$. Consider further an encoder $\Phi \in \mathbb{R}^{m \times N}$ which has the $(2k, \delta_{2k})$-RIP, with $\delta_{2k} < \sqrt{2}-1$, the respective measurement vector $y = \Phi x \in \mathbb{R}^m$, and the iteratively re-weighted ℓ_1-minimizer $x^* := \Delta_{1rew}(y)$. If for all $i \in \mathrm{supp}(x_{[k]})$

$$|x_i| > 9.6 \frac{\sqrt{1+\delta_{2k}}}{1-(\sqrt{2}+1)\delta_{2k}} \left(1 + \frac{\kappa_p}{\sqrt{k}}\right) \eta =: r_{1rew}, \tag{133}$$

then $\mathrm{supp}(x_{[k]}) \subset \mathrm{supp}(x^*)$.

Proof. A proof of this theorem is given in [2, Theorem 2].

Here, the noisy signal should have the k largest entries in absolute value above $r_{1\mathrm{rew}}$ in order to guarantee support identification.

5.2 Support Identification Stability in a Class of Sparse Vectors Affected by Bounded Noise

Let us introduce for $r > \eta > 0$, $1 \leq k < m$, and $1 \leq p \leq 2$, the class of *sparse vectors affected by bounded noise*,

$$\mathcal{S}^p_{\eta,k,r} := \left\{ x \in \mathbb{R}^N \middle| \#S_r(x) \leq k \text{ and } \sum_{i \in (S_r(x))^c} |x_i|^p \leq \eta^p \right\}, \tag{134}$$

where $S_r(x) := \{i \in \{1,\ldots,N\} |\, |x_i| > r\}$ is the index support of the large entries exceeding in absolute value the threshold r. This class contains all vectors for which at most $1 \leq k < m$ large entries exceed the threshold r in absolute value, while the p-norm of the other entries stays below a certain noise level. Notice that vectors $x \in \mathcal{S}^p_{\eta,k,r}$ can be naturally decomposed in the noiseless (relevant) part $\bar{x} = x_{S_r(x)}$ and the noise $n = x_{S_r^c(x)}$. In this section, we present results in terms of support discrepancy once we consider two elements of the class $\mathcal{S}^p_{\eta,k,r}$, having the same measurements.

Theorem 18. *Let $\Phi \in \mathbb{R}^{m \times N}$ have the $(2k, \gamma_{2k})$-NSP, for $\gamma_{2k} < 1$, $1 \leq p \leq 2$, and $x, x' \in \mathcal{S}^p_{\eta,k,r}$ such that $\Phi x = \Phi x'$, and $0 \leq \eta < r$. Then*

$$\#(S_r(x) \Delta S_r(x')) \leq \frac{(2\gamma_{2k}\kappa_p \eta)^p}{(r-\eta)^p}. \tag{135}$$

(Here we denote by "Δ" the set symmetric difference, not to be confused with the previously introduced symbol of a generic decoder.) If additionally

$$r > \eta(1 + 2\gamma_{2k}\kappa_p) =: r_S, \tag{136}$$

then $S_r(x) = S_r(x')$.

Proof. As $\Phi x = \Phi x'$, then $(x - x') \in \ker(\Phi)$. By the $(2k, \gamma_{2k})$-NSP, Hölder's inequality, and the triangle inequality we have

$$\begin{aligned}\left\|(x-x')_{S_r(x)\cup S_r(x')}\right\|_{\ell_p} &\le \left\|(x-x')_{S_r(x)\cup S_r(x')}\right\|_{\ell_1} \le \gamma_{2k}\left\|(x-x')_{(S_r(x)\cup S_r(x'))^c}\right\|_{\ell_1}\\ &\le \gamma_{2k}\kappa_p\left\|(x-x')_{(S_r(x)\cup S_r(x'))^c}\right\|_{\ell_p} \le 2\gamma_{2k}\kappa_p\eta. \end{aligned} \tag{137}$$

Now we estimate the symmetric difference of the supports of the large entries of x and x' in absolute value as follows: if $i \in S_r(x)\Delta S_r(x')$, then either $|x_i| > r$ and $|x'_i| \le \eta$ or $|x_i| \le \eta$ and $|x'_i| > r$. This implies that $|x'_i - x_i| > (r-\eta)$. Thus we have $\left\|(x-x')_{S_r(x)\Delta S_r(x')}\right\|_{\ell_p}^p \ge (\#(S_r(x)\Delta S_r(x')))\,(r-\eta)^p$. Together with the non-negativity of $\left\|(x-x')_{S_r(x)\cap S_r(x')}\right\|_{\ell_p}$, we obtain the chain of inequalities

$$\begin{aligned}(2\gamma_{2k}\kappa_p\eta)^p &\ge \left\|(x-x')_{S_r(x)\cup S_r(x')}\right\|_{\ell_p}^p \ge \left\|(x-x')_{S_r(x)\cap S_r(x')}\right\|_{\ell_p}^p + \left\|(x-x')_{S_r(x)\Delta S_r(x')}\right\|_{\ell_p}^p\\ &\ge \left(\#(S_r(x)\Delta S_r(x'))\right)(r-\eta)^p,\end{aligned}$$

and therefore we obtain (135). Notice now that (135) and (136) implies $\mathbb{N} \ni \#(S_r(x)\Delta S_r(x')) < 1$ and $S_r(x)\Delta S_r(x') = \emptyset$. $\square$

Remark 7. One additional implication of this latter theorem is that we can give a bound on the difference of x and x' restricted to the relevant entries. Indeed, in case of unique identification of the relevant entries, i.e., $\Lambda := S_r(x) = S_r(x')$ we obtain, by the inequality (137), that

$$\left\|(x-x')_\Lambda\right\|_{\ell_1} \le 2\gamma_k\kappa_p\eta. \tag{138}$$

Notice that we replaced γ_{2k} by $\gamma_k \le \gamma_{2k}$, because now $\#\Lambda \le k$.

Unfortunately, we are not able to provide the *necessity of the gap conditions* (129), (133), (136) for successful support recovery, simply because we lack optimal deterministic error bounds in general: one way of producing a lower bound would be to construct for each algorithm a counterexample, for which a certain gap condition is violated and recovery of support fails. Since most of the algorithms we shall illustrate below are iterative, it is likely extremely difficult to provide such explicit counterexamples. Therefore, we limit ourselves here to discuss the discrepancies of r_1 and r_S and of $r_{1\text{rew}}$ and r_S. We shall see in the numerical experiments that the *sufficient gap conditions* (129), (133), (136) nevertheless provide actual indications of performance of the algorithms.

The gap between the two thresholds r_1, r_S is given by

$$r_1 - r_S = \left(2\left(\frac{1+\gamma_k}{1-\gamma_k} - \gamma_{2k}\right)\kappa_p(N,k) - 1\right)\eta.$$

As $\gamma_{2k} < 1 < \frac{1+\gamma_k}{1-\gamma_k}$ and $\kappa_p(N,k)$ is very large for $N \gg k$ and $p > 1$, this *positive* gap is actually very large, for $N \gg 1$.

The gap between the two thresholds $r_{1\text{rew}}$, r_S is given by

$$r_{1\text{rew}} - r_S = \left(9.6\frac{\sqrt{1+\delta_{2k}}}{1-(\sqrt{2}+1)\delta_{2k}}\left(1+\frac{\kappa_p}{\sqrt{k}}\right) - (1+2\gamma_{2k}\kappa_p)\right)\eta.$$

Following, for example, the arguments in [23], we know that a matrix Φ having the $(2k,\delta_{2k})$-RIP has also the $(2k,\gamma_{2k})$-NSP with $\gamma_{2k} = \frac{\sqrt{2}\delta_{2k}}{1-(\sqrt{2}+1)\delta_{2k}}$, which, substituted into the above equation, yields

$$r_{1\text{rew}} - r_S = \frac{\left(\frac{9.6\sqrt{1+\delta_{2k}}}{\sqrt{k}} - 2\sqrt{2}\delta_{2k}\right)\kappa_p + \left[9.6\sqrt{1+\delta_{2k}} - (1-(\sqrt{2}+1)\delta_{2k})\right]}{1-(\sqrt{2}+1)\delta_{2k}}\,\eta.$$

Since $0 < \delta_{2k} < \sqrt{2}-1$, we have $0 < 1-(\sqrt{2}+1)\delta_{2k} < 1$, and therefore the denominator and the right summand in the numerator are positive. The left summand of the numerator is positive and very large as soon as $k < \left(\frac{9.6\sqrt{1+\delta_{2k}}}{2\sqrt{2}\delta_{2k}}\right)^2$. Thus, even in the limiting scenario where $\delta_{2k} \approx \sqrt{2}-1$, we still have $k \leq 94$, which may be considered sufficient for a wide range of applications. A more sophisticated estimate of the above term can actually reveal even less restrictive bounds on k. Thus, in general, since k and δ_{2k} are small, also the left summand is positive. We conclude again that the gap is large for $N \gg k$ and $p > 1$.

Unfortunately, the discrepancies $r_1 - r_S \gg 0$ and $r_{1\text{rew}} - r_S \gg 0$ cannot be amended because in general the ℓ_1-minimization decoder Δ and the iteratively re-weighted ℓ_1-minimization decoder $\Delta_{1\text{rew}}$ have *not* the property of decoding a vector in the class $\mathcal{S}^p_{\eta,k,r}$, even if the original vector x belonged to it, i.e., in general the implication

$$x \in \mathcal{S}^p_{\eta,k,r} \Rightarrow \{\Delta(\Phi x), \Delta_{1\text{rew}}(Ax)\} \ni x^* \in \mathcal{S}^p_{\eta,k,r} \tag{139}$$

does *not* hold for these decoders. These ineliminable limitations of Δ and $\Delta_{1\text{rew}}$ can be verified, e.g., in a specific numerical counterexample in [2, Figure 4], where (139) does not hold for the ℓ_1-minimizer.

5.3 Non-convex Methods for Enhanced Support Identification Properties

To overcome the shortcomings of methods based exclusively on ℓ_1-minimizations in

1. damping the noise-folding, and consequently in
2. having a stable support recovery,

in this section, we summarize the design and the properties of two decoding procedures, introduced in [2], with output in $\mathcal{S}^p_{\eta,k,r}$, which consequently allow us to have both these very desirable properties.

5.3.1 Properties of the Regularized Selective p-Potential Functional (SLP)

Let us first introduce the following functional.

Definition 4 (Regularized selective p-potential). We define the *regularized truncated p-power function* $W_r^{p,\varepsilon}: \mathbb{R} \to \mathbb{R}_0^+$ by

$$W_r^{p,\varepsilon}(t) = \begin{cases} t^p & 0 \leq t < r-\varepsilon, \\ \pi_p(t) & r-\varepsilon \leq t \leq r+\varepsilon, \\ r^p & t > r+\varepsilon, \end{cases} \quad t \geq 0, \tag{140}$$

where $0 < \varepsilon < r$, and $\pi_p(t)$ is the third degree interpolating polynomial

$$\pi_p(t) := A(t-s_2)^3 + B(t-s_2)^2 + C,$$

and $C=\mu_3$, $B=\frac{\mu_1}{s_2-s_1}-\frac{3(\mu_3-\mu_2)}{(s_2-s_1)^2}$, $A=\frac{\mu_1}{3(s_2-s_1)^2}+\frac{2B}{3(s_2-s_1)}$, where $s_1=(r-\varepsilon)$, $s_2=(r+\varepsilon)$, $\mu_1=p(r-\varepsilon)^{p-1}$, $\mu_2=(r-\varepsilon)^p$, and $\mu_3=r^p$. Moreover, we set $W_r^{p,\varepsilon}(t)=W_r^{p,\varepsilon}(-t)$ for $t<0$. We call the functional $\mathcal{SP}_r^{p,\varepsilon}:\mathbb{R}^N\to\mathbb{R}_0^+$,

$$\mathcal{SP}_r^{p,\varepsilon}(x)=\sum_{j=1}^N W_r^{p,\varepsilon}(x_j),\quad r>0,\quad 1\le p\le 2, \tag{141}$$

the *regularized selective p-potential (SP) functional*.

The graphs of $W_r^{p,0}$ and $W_r^{p,\varepsilon}$ are shown in Figure 13 for $p=2$, $r=1$, and $\varepsilon=0.4$, see [3, 32] for related literature and further details in statistical signal processing.

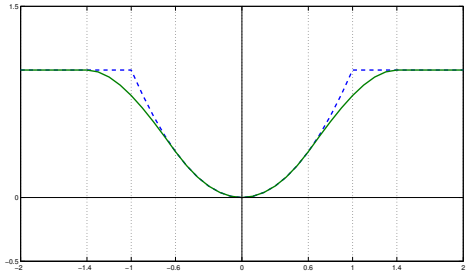


Fig. 13 Truncated quadratic potential $W_1^{2,0}$ (dashed) and its regularization $W_1^{2,0.4}$.

Theorem 19. *Let $\Phi\in\mathbb{R}^{m\times N}$ have the $(2k,\gamma_{2k})-$NSP, with $\gamma_{2k}<1$, and $1\le p\le 2$. Furthermore, we assume $x\in\mathcal{S}^p_{\eta,k,r+\varepsilon}$, for $\varepsilon>0$, $0<\eta<r+\varepsilon$, with the property of having the minimal $\#S_{r+\varepsilon}(x)$ within $\mathcal{F}_\Phi(y)$, where $y=\Phi x$ is its associated measurement vector, i.e.,*

$$\#S_{r+\varepsilon}(x)\le\#S_{r+\varepsilon}(z)\ \text{for all}\ z\in\mathcal{F}_\Phi(y). \tag{142}$$

If x^* is such that

$$\mathcal{SP}_r^{p,\varepsilon}(x^*)\le\mathcal{SP}_r^{p,\varepsilon}(x), \tag{143}$$

and

$$|x_i^*|<r-\varepsilon, \tag{144}$$

for all $i\in(S_{r+\varepsilon}(x^*))^c$, then also $x^*\in\mathcal{S}^p_{\eta,k,r+\varepsilon}$, implying noise-folding damping. Moreover, we have the support stability property

$$\#(S_{r+\varepsilon}(x)\Delta S_{r+\varepsilon}(x^*))\le\frac{(2\gamma_{2k}\kappa_p\eta)^p}{(r+\varepsilon-\eta)^p}. \tag{145}$$

Proof. As this algorithm does not perfom well for N large and it provides almost exclusively theoretical insights, we do not include the proof of this theorem and we refer to [2, Theorem 4] for the details. □

We may want to stress the meaning of this result. The potential $\mathcal{SP}_r^{p,\varepsilon}$ is somehow "selective". If a component is large, it simply counts it, if instead it is relatively small it damps it, and encodes it as noise to be filtered.

Remark 8. Let us comment the assumptions of the latter result.

(i) The assumption that x is actually the vector with minimal essential support $S_r(x)$ among the feasible vectors in $\mathcal{F}_\Phi(y)$ corresponds to the request of being the "simplest" explanation to the data;

(ii) The best candidate x^* to fulfill condition (143) would be actually

$$x^* := \operatorname*{arg\,min}_{z \in \mathcal{F}_\Phi(y)} \mathcal{SP}_r^{p,\varepsilon}(z) \tag{146}$$

because this will make (143) automatically true, whichever x is. However (146) is a highly nonconvex problem whose solution is in general NP-hard [1]. The way we will circumvent this drawback is to employ an iterative algorithm, which we call *selective least p-powers (SLP)*, to compute x^*, performing a local minimization of $\mathcal{SP}_r^{p,\varepsilon}$ in $\mathcal{F}_\Phi(y)$ around a given starting vector x_0, see [3, Section 3.3] and the following sections on that paper, or the detailed description in [2]. Ideally, the best choice for x_0 would be x itself, so that (143) may be fulfilled. As we do not dispose yet of the original vector x, a heuristic rule, which we will show to be very robust in our numerical simulations, is to choose $x_0 = \Delta(y) \approx x$. The reasonable hope is that actually $\mathcal{SP}_r^{p,\varepsilon}(x^*) \leq \mathcal{SP}_r^{p,\varepsilon}(\Delta(y)) \approx \mathcal{SP}_r^{r,\varepsilon}(x)$;

(iii) The assumption that the outcome x^* of the algorithm has additionally the property $|x_i^*| < r - \varepsilon$, for all $i \in (S_{r+\varepsilon}(x^*))^c$ is justified by observing that in the actual implementation x^* will be the result of a thresholding operation, i.e., $x_i^* = \mathbb{S}_p^\mu(\xi_i)$, for $i \in (S_{r+\varepsilon}(x^*))^c$, where $\mathbb{S}_p^\mu$ is defined as in [3, Formula 3.36]. The particularly steep shape of the thresholding functions $\mathbb{S}_p^\mu$ in the interval $[r-\varepsilon, r+\varepsilon]$, especially for $p = 2$, see [3, Figure 3.3 (c)], makes it highly unlikely for ε sufficiently small that $r - \varepsilon \leq |x_i^*|$ for $i \in (S_{r+\varepsilon}(x^*))^c$.

5.3.2 Properties of Iterative Hard Thresholding (IHT-λ)

As already mentioned above, the numerical realization of the algorithm SLP from [3] turns out to be computationally demanding as soon as the dimension N gets large. In the following, we show that the method IHT-λ, which is introduced in Section 2.3.2, shows similar support identification properties as SLP while being very efficient in terms of computational time. It is also somehow implementing a selectivity principle, distinguising signal from noise. In Theorem 20, we discuss under which sufficient conditions this method is able to exactly identify the support of the relevant entries of the original vector x.

Theorem 20. *Assume $\Phi \in \mathbb{R}^{m \times N}$ to have the $(2k, \delta_{2k})$-RIP, with $\delta_{2k} < 1$, $\|\Phi\| \leq 1$, and define $\beta(\Phi) > 0$ as in* (111). *Let $x \in \mathcal{S}_{\eta,k,r}^p$, for a fixed $1 \leq p \leq 2$, and $y = \Phi x$ the respective measurements. Assume further*

$$r > \eta \left(1 + \frac{1}{1-\delta_{2k}} \left(1 + \frac{1}{\beta(\Phi)}\right)\right), \tag{147}$$

and define λ such that

$$\eta < \sqrt{\lambda} < \frac{r - \frac{\eta}{1-\delta_{2k}}}{1 + \frac{1}{(1-\delta_{2k})\beta(\Phi)}}. \tag{148}$$

Let x^h be the limit of the sequence generated by Algorithm IHT-λ, and we assume

$$\mathcal{J}_0(x^h) \leq \mathcal{J}_0(x_{S_r(x)}). \tag{149}$$

Then $\Lambda := S_r(x) = \operatorname{supp}(x^h)$, and it holds

$$|x_i - x_i^h| < r - \sqrt{\lambda}, \text{ for all } i \in \Lambda. \tag{150}$$

Proof. Assume $\#\mathrm{supp}(x^{\mathrm{h}}) > \#S_r(x) = k$. By (149), we have that

$$\begin{aligned} 0 < \#\mathrm{supp}(x^{\mathrm{h}}) - \#\mathrm{supp}(x_{S_r(x)}) = \#\mathrm{supp}(x^{\mathrm{h}}) - \#S_r(x) \leq \frac{1}{\lambda}\left(\left\|\Phi\left(x_{S_r(x)}\right) - y\right\|_{\ell_2}^2 - \left\|\Phi x^{\mathrm{h}} - y\right\|_{\ell_2}^2\right) \\ \leq \frac{1}{\lambda}\left\|\Phi\left(x_{S_r(x)}\right) - y\right\|_{\ell_2}^2 = \frac{1}{\lambda}\left\|\Phi\left(x_{(S_r(x))^c}\right)\right\|_{\ell_2}^2 \leq \frac{1}{\lambda}\|\Phi\|^2\left\|x_{(S_r(x))^c}\right\|_{\ell_2}^2 \leq \frac{\eta^2}{\lambda} < 1, \end{aligned}$$

where the last inequality follows by (148). Since $\left(\#\mathrm{supp}\left(x_{S_r(x)}\right) - \#\mathrm{supp}\, x^{\mathrm{h}}\right) \in \mathbb{N}$, the upper inequality yields to a contradiction. Thus $\#\mathrm{supp}(x^{\mathrm{h}}) \leq \#S_r(x) = k$ and therefore x^{h} and $x_{S_r(x)}$ are both k-sparse, and $\left(x^{\mathrm{h}} - x_{S_r(x)}\right)$ is $2k$-sparse. Under our assumptions we can apply Theorem 12 to obtain

$$\left\|\Phi x^{\mathrm{h}} - y\right\|_{\ell_2} \leq \frac{\sqrt{\lambda}}{\beta(\Phi)}. \tag{151}$$

In addition to this latter estimate, we use the RIP, the sparsity of $x^{\mathrm{h}} - x_{S_r(x)}$, and (148) to obtain for all $i \in \{1,\ldots,N\}$ that

$$\begin{aligned} \left|\left(x_{S_r(x)}\right)_i - x_i^{\mathrm{h}}\right| \leq \left\|\left(x_{S_r(x)}\right) - x^{\mathrm{h}}\right\|_{\ell_2} \leq \frac{\left\|\Phi(x_{S_r(x)} - x^{\mathrm{h}})\right\|_{\ell_2}}{1-\delta_{2k}} \leq \frac{\left\|\Phi(x - x^{\mathrm{h}})\right\|_{\ell_2} + \left\|\Phi\left(x_{(S_r(x))^c}\right)\right\|_{\ell_2}}{1-\delta_{2k}} \\ \leq \frac{\left\|y - \Phi x^{\mathrm{h}}\right\|_{\ell_2} + \left\|\Phi\left(x_{(S_r(x))^c}\right)\right\|_{\ell_2}}{1-\delta_{2k}} \leq \frac{\sqrt{\lambda}}{\beta(\Phi)(1-\delta_{2k})} + \frac{\eta}{1-\delta_{2k}} < r - \sqrt{\lambda}. \end{aligned}$$

Assume now that there is $\tilde{i} \in \mathbb{N}$ such that $\tilde{i} \in S_r(x)$ and $\tilde{i} \notin \mathrm{supp}(x^{\mathrm{h}})$. But then we would also have $|x_{\tilde{i}} - x_{\tilde{i}}^{\mathrm{h}}| = |x_{\tilde{i}}| > r$, which leads to a contradiction. Thus, $S_r(x) \subset \mathrm{supp}(x^{\mathrm{h}})$, which together with $\#\mathrm{supp}(x^{\mathrm{h}}) \leq \#S_r(x)$ conclude the proof. □

Remark 9. Let us discuss some of the assumptions and implications of this latter result.

(i) Since iterative hard thresholding only computes a local minimizer of $\mathcal{J}_0$, condition (149) may not be always fulfilled for any given initial iteration x^0. Similarly to the argument in Remark 8 (ii), using the ℓ_1-minimizer as the starting point x^0, or equivalently choosing the vector x^0 as composed of the entries of $\Delta(\Phi x)$ exceeding $\sqrt{\lambda}$ in absolute value, we may allow us to approach a local minimizer which fulfills (149).

(ii) Condition (147) is comparable to the one derived in (136). If Φ is "well-conditioned", i.e., we have that $(1-\delta_{2k}) \sim 1$, and $\beta(\Phi) \sim 1$, then

$$1 + \frac{1}{1-\delta_{2k}}\left(1 + \frac{1}{\beta(A)}\right) \sim 3.$$

We remind that besides the exact identification of the support Λ of the original signal $\bar{x}$, which is provided by Theorem 20, the goal in this section is also to find an accurate reconstruction of its relevant entries. In this sense, the relatively poor error estimate (150) is not satisfactory. The reason why we cannot obtain an estimate as good as (138) in Remark 7 is that the assumptions of Theorem 18, i.e., the conditions $x^{\mathrm{h}} \in \mathcal{S}^p_{\eta,k,r}$, and $\Phi x = \Phi x^{\mathrm{h}}$, are in general not fulfilled. To obtain a modification x' of x^{h}, such that these conditions are satisfied, an additional correction is necessary. It is a natural approach to determine the vector x' as the solution of

$$\begin{aligned} \min_{z\in\mathbb{R}^N} \quad & \|\Phi z - y\|_{\ell_2}^2 \\ \text{s.t.} \quad & \|z_{\Lambda^c}\|_{\ell_p} \le \eta, \\ & |z_i| \ge r, \text{ for all } i \in \Lambda, \end{aligned} \tag{152}$$

being $\Lambda = S_r(x) = \mathrm{supp}(x^{\mathrm{h}})$ the support already identified. Since the original signal x fulfills $\Phi x - y = 0$, and $x \in \mathcal{S}^p_{\eta,k,r}$, it is actually a solution of problem (152). Thus, we conclude that for any minimizer x' of problem (152) the objective function equals zero, thus $\Phi x = \Phi x'$ and, simultaneously, $x' \in \mathcal{S}^p_{\eta,k,r}$. The optimization (152) is in general nonconvex, but we can easily recast it in an equivalent convex one: Since $|x_i - x_i^{\mathrm{h}}| < r - \sqrt{\lambda}$, and $|x_i| > r$, we know that the relevant entries of x and x^{h} have the same sign. Since we are searching for solutions which are close to x, the second inequality constraint becomes $\mathrm{sign}(x_i^{\mathrm{h}})z_i \ge r$, for all $i \in \Lambda$. Together with the equivalence of ℓ_2- and ℓ_p-norm, we rewrite problem (152) as

$$\begin{aligned} \min_{z\in\mathbb{R}^N} \quad & \frac{1}{2} z^*(\Phi^*\Phi)z - y^*\Phi z \\ \text{s.t.} \quad & z^* P_0 z - (N-k)^{1-\frac{2}{p}}\eta^2 \le 0, \\ & z^* P_j z - (\mathrm{sign}(x^{\mathrm{h}}_{i_j})e_{i_j})^* z + r \le 0, \\ & \qquad \text{for all } i_j \in \Lambda \ , \ j = 1, \ldots, \#\Lambda, \end{aligned} \tag{153}$$

where $P_0 \in \mathbb{R}^{N\times N}$ is defined componentwise by

$$(P_0)_{r,s} := \begin{cases} 1 & \text{if } r = s \in \Lambda \\ 0 & \text{else} \end{cases},$$

and $P_j = 0$, $j = 1, \ldots \#\Lambda$. Since $\Phi^*\Phi$, P_0, and P_j, $j = 1, \ldots, \#\Lambda$, are semi-definite, problem (153) is a *convex quadratically constrained quadratic program* (QCQP) which can be efficiently solved by standard methods, e.g., interior point methods [56]. Since we combine here three very efficient methods (ℓ_1-minimization, IHT-λ, and a QCQP), the resulting procedure is much faster than the computation of SLP while, as we will show in the numerics, keeping similar support identification properties.

5.4 Numerical Results

The following numerical simulations provide empirical confirmation of the theoretical observations in this section. In particular, we want to show that the methods ℓ_1+SLP (SLP with ℓ_1-minimization warm-up) and ℓ_1+IHT-λ (IHT-λ with ℓ_1-minimization warm-up and final convex correction step) are very robust and provide a significantly enhanced rate of recovery of the support of the unknown sparse vector as well as a better accuracy in approximating its large entries, with respect to the sole ℓ_1-minimization or its re-weighted version, whenever limiting noise, i.e., $\eta \approx r$, is present on the signal. If we refer to "SLP", we mean the SLP method initialized by 0.

We also consider as one of the test methods BPDN (compare (23)), with the parameter $\eta = \sigma^2(m + 2\sqrt{2m})$. The stability parameter a in iteratively re-weighted ℓ_1-minimization (IRWℓ_1), which avoids the denominator to be zero in the weight updating rule of IRWℓ_1 seems not to have a strong influence, and it

is set to 0.1 in our experiments. We executed 8 iterations of IRWℓ_1 as a reasonable compromise between computational effort and accuracy.
In order to fulfill the assumptions of our theoretical results, we use for the numerical experiments random matrices, satisfying the RIP with optimal constants with high probability. In particular all tests presented in this section are realized with column-wise normalized i.i.d. Gaussian encoding matrices.
In this section we show a few representative results. More detailed numerical results can be found in [2]. All tests were implemented and run in Matlab R2013b in combination with CVX [35, 66], to solve ℓ_1-minimization and BPDN, its iteratively re-weighted version, and the QCQP.

5.4.1 Empirical Statistics

We fix the parameters in order to have the most coherent data to be analyzed; in particular, we set $N = 100$, $m = 40$, $r = 0.8$, $k = 1,\dots,7$, and $\eta = 0.75$. The vector n is composed of random entries with normal distribution and then it is rescaled in order to have $\|n\|_{\ell_2} = \eta$. Figures 14, and 15 report the results obtained considering 30 different i.i.d. Gaussian encoding matrices. In the following, we use x^* generically for the decoded vector of any method. In Figure 14, we report the histogram of the mean-value of the errors on the

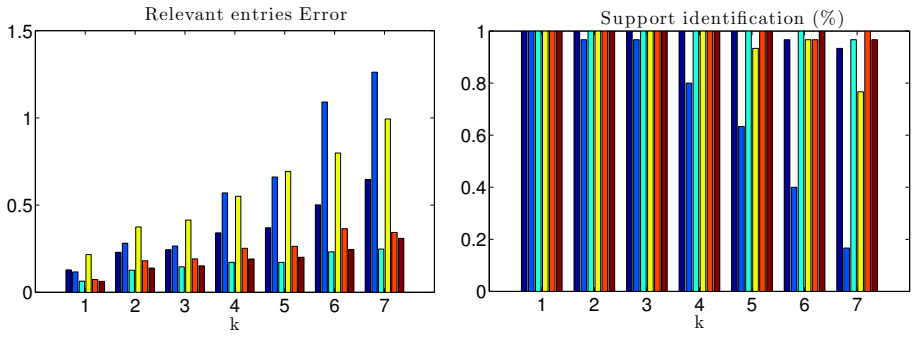


Fig. 14 The columns refer to the different results of ℓ_1-minimization (dark blue), SLP (blue), ℓ_1+SLP (cyan), BPDN (yellow), IRWℓ_1 (orange), and ℓ_1+IHT-λ (brown). The subfigures represent the error on the relevant entries and the support identification property by knowledge of k.

relevant entries: the quantities on the left subfigure are computed as the mean values of $\left\|x_{S_r(x)} - x^*_{S_r(x)}\right\|_{\ell_2}$ where we suppose to know the k largest entries of the original signal. The right subfigure shows how often the k largest entries of x^* coincided with $S_r(x)$. Notice that there might be entries below the threshold r among the k largest entries of x^*. We conclude that, knowing the number of large entries, IRWℓ_1, ℓ_1-minimization, ℓ_1+SLP, and ℓ_1+IHT-λ recover the support with nearly 100% success. In addition, ℓ_1+SLP approximates best the magnitudes of the relevant entries.

In Figure 15 we compute again the mean-value of the relevant entries, but this time without the knowledge of k but the knowledge of r and therefore $S_r(x^*)$: the quantities on the left subfigure are the mean values of $\left\|x_{S_r(x^*)} - x^*_{S_r(x^*)}\right\|_{\ell_2}$. In the right subfigure we attribute a positive match in case $S_r(x^*) = S_r(x)$ so that the relevant entries of x^* coincide with the ones of the original signal. By our theory, we expect ℓ_1+SLP and

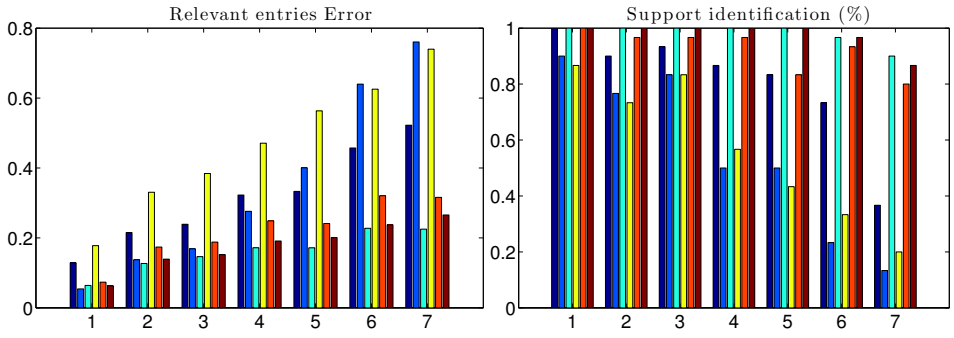


Fig. 15 The subfigures represent the error on the relevant entries and the support identification property by knowledge of r. For more details on the displayed data we refer to the caption of Figure 14. The results were obtained by Gaussian matrices.

ℓ_1+IHT-λ to produce a high rate of success of correctly recovered support. Actually this is confirmed by the experiments: Both methods do a very accurate recovery, as it gives us almost always 100% of the correct result while the other methods perform worse.

We conclude that the a priori knowledge of the number of non-zero elements k significantly ameliorates the ability to identify the support correctly, even more for ℓ_1-based methods. If this knowledge is missing and only the threshold r is known, the performance of ℓ_1 based methods stays behind the performance of the introduced non-convex methods.

5.4.2 Phase Transition Diagrams

To give an even stronger support of the results in the previous paragraph, we extended the results of Figure 15 to a wider range of m and k. In Figure 16, we present phase transition diagrams of success rates in support recovery for ℓ_1-minimization, IRWℓ_1, ℓ_1+SLP, and ℓ_1+IHT-λ in presence of nearly maximally allowed noise, i.e., $0.8 = r > \eta = 0.75$.

To produce phase transition diagrams, we varied the dimension of the measurement vector $m = 1,\dots,N$ with $N = 100$, and solved 20 different problems for all the admissible $k = \#S_r(x) = 1,\dots,m$. We colored black all the points (m,k), with $k \leq m$, which reported 100% of correct support identification, and gradually we reduce the tone up to white for the 0% result. The level bound of 50% and 90% is highlighted by a magenta and red line respectively. A visual comparison of the corresponding phase transitions confirms our previous expectations. In particular, ℓ_1+SLP and ℓ_1+IHT-λ very significantly outperform ℓ_1-minimization in terms of correct support recovery. The difference of both methods towards IRWℓ_1 is less significant but still important. In Figure 17, we compare the level bounds of 50% and 90% among the four different methods. Observe that the 90% probability bound indicates the largest positive region for ℓ_1+IHT-λ, followed by ℓ_1+SLP, and only eventually by IRWℓ_1, while the bounds are much closer to each other in the case of the 50% bound. Thus, surprisingly, ℓ_1+IHT-λ works in practice even better than ℓ_1+SLP for some range of m, and offers the most stable support recovery results.

Acknowledgements The authors acknowledge the support of the project "Sparse Reconstruction and Compressive Sensing for Remote Sensing and Earth Observation" funded by Munich Aerospace.

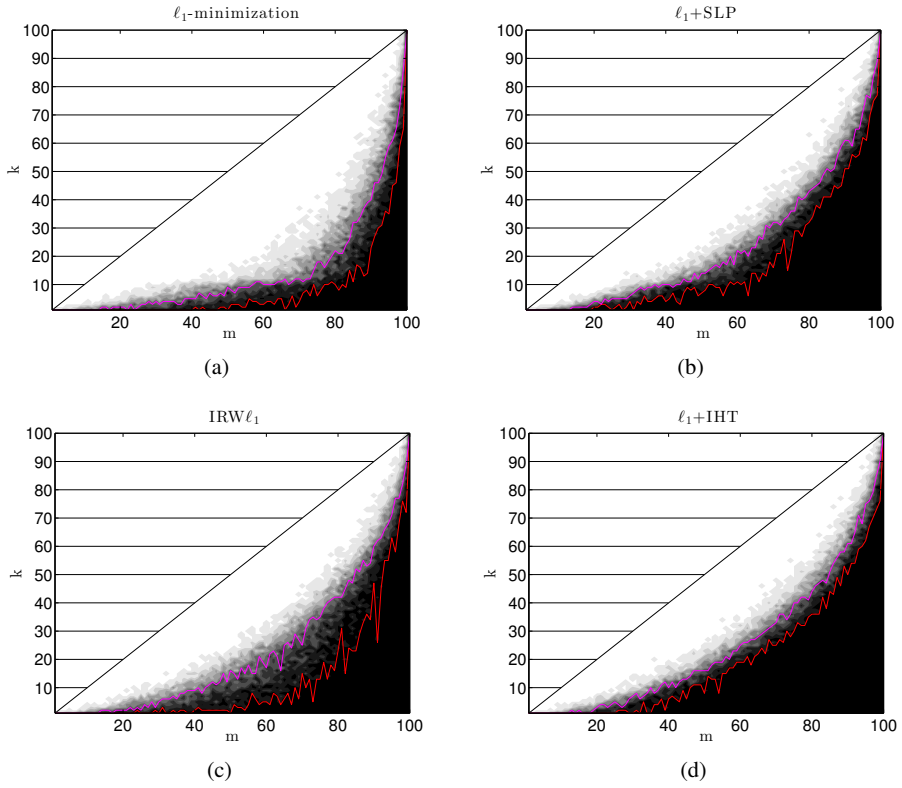


Fig. 16 Phase transition diagrams. The black area represents the couple (m,k) for which we had 100% of support recovery. The results of (a) ℓ_1-minimization, (b) ℓ_1+SLP, (c) IRWℓ_1, and (d) ℓ_1+IHT-λ are reported. Note that the area for $k > m$ is not admissible. The red line shows the level bound of 90% of support recovery, and the magenta line 50% respectively.

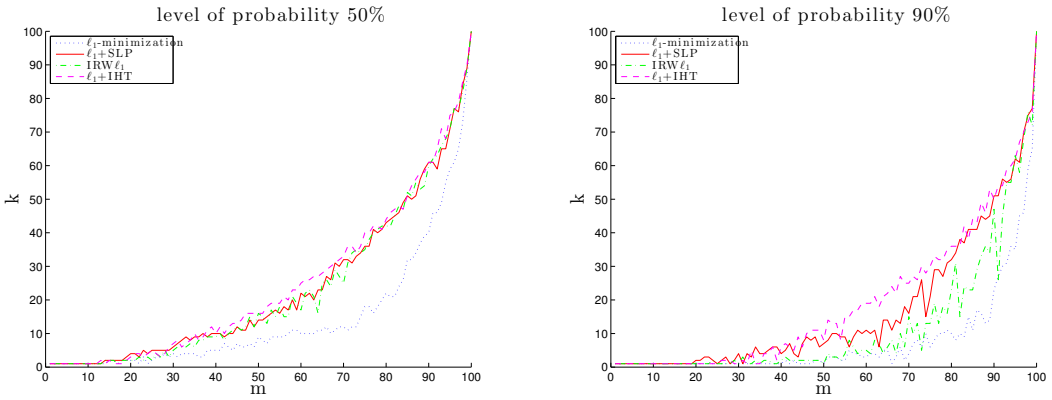
Appendix

A Conjugate Gradient Methods

Since in the literature a wide variety of Conjugate Gradient methods in terms of formulation and notation appears, we want to present some Conjugate Gradient methods to be used in this Chapter and state corresponding convergence results. Thus, we also introduce a proper consistent notation to avoid confusion.

Classical Conjugate Gradient Method (CG)

The CG method was originally proposed by Stiefel and Hestenes in [39] and generalized to complex systems in [43]. Let the matrix $A \in \mathbb{C}^{N\times N}$ be Hermitian and positive definite, then the CG method solves the linear equation $Ax = y$ or equivalently the minimization problem



(a) (b)

Fig. 17 Comparison of phase transition diagrams for ℓ_1-minimization (dark blue, dotted), ℓ_1+SLP (red), IRWℓ_1 (green, dash-dotted), and ℓ_1+IHT (magenta, dashed). The level bound of 50% and 90% as it is displayed in Figure 16 is compared in (a) and (b) respectively.

$$\underset{x\in\mathbb{C}^N}{\arg\min}\left(F(x) := \frac{1}{2}x^*Ax - x^*b\right).$$

The algorithm is designed to iteratively compute the minimum x^i of F on the Krylov subspace $V_i := \text{span}\{y, Ay, \ldots, A^{i-1}y\} \subset \mathbb{C}^N$. The solution is found after N iterations in exact precision since $V_N = \mathbb{C}^N$.

Algorithm 8 Conjugate Gradient (CG) method

Input: initial vector $x^0 \in \mathbb{C}^N$, matrix $A \in \mathbb{C}^{N\times N}$, given vector $y \in \mathbb{C}^N$ and optionally a desired accuracy δ.

Set $r^0 = p^0 = y - Ax^0$ and $i = 0$
while $r^i \neq 0$ (or $\left\|r^i\right\|_{\ell_2} > \delta$) **do**
  $a_i = \langle r^i, p^i\rangle_{\ell_2} / \langle Ap^i, p^i\rangle_{\ell_2}$
  $x^{i+1} = x^i + a_i p^i$
  $r^{i+1} = y - Ax^{i+1}$
  $b_{i+1} = \langle Ap^i, r^{i+1}\rangle_{\ell_2} / \langle Ap^i, p^i\rangle_{\ell_2}$
  $p^{i+1} = r^{i+1} - b_{i+1}p^i$
  $i = i + 1$
end while

Roughly speaking, CG is iteratively searching for a minimum of the functional F along conjugate directions p^i with respect to A, i.e., $(p^i)^*Ap^j = 0$, $i \neq j$. Thus, in step $i+1$ of CG the new iterate x^{i+1} is found by minimizing $F(x^i + a_i p^i)$ with respect to the scalar $a_i \in \mathbb{R}$ along the aforementioned search direction p^i. Since we perform a minimization in each iteration, in particular, we have the monotonicity of the iterates expressed by the inequality $F(x^{i+1}) \leq F(x^i)$.

The following theorem establishes the convergence and the convergence rate of CG.

Theorem 21 ([62, Theorem 4.12]). *Let the matrix A be Hermitian and positive definite. The Algorithm CG converges to the solution of the system $Ax = y$ after at most N steps. Moreover, the error $x^i - x$ is such that*

$$\left\| A^{\frac{1}{2}}(x^i - x) \right\|_{\ell_2} \leq \frac{2c_A^i}{1 + c_A^{2i}} \left\| A^{\frac{1}{2}}(x^0 - x) \right\|_{\ell_2}, \quad \text{with } c_A = \frac{\sqrt{\kappa_A} - 1}{\sqrt{\kappa_A} + 1},$$

where $\kappa_A = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$ is the condition number of the matrix A and $\sigma_{\max}(A)$ (resp. $\sigma_{\min}(A)$) is the largest (resp. smallest) singular value of A.

Remark 10. Theorem 21 is slightly modified with respect to the formulation in [62]. There, the matrix A is considered to be symmetric instead of being Hermitian. However, in the complex case, the proof can be performed similarly by replacing the transpose by the conjugate transpose.

Modified Conjugate Gradient Method (MCG)

In Section 2.2.1.5, we are interested in a vector which solves the weighted least-squares problem

$$\hat{x} = \underset{x \in \mathcal{F}_\Phi(y)}{\arg\min} \|x\|_{\ell_2(w)},$$

given $\Phi \in \mathbb{C}^{m \times N}$ with $m \leq N$. The minimizer $\hat{x}$ is given explicitly by the (weighted) Moore-Penrose pseudo-inverse

$$\hat{x} = D\Phi^*(\Phi D \Phi^*)^{-1} y,$$

where $D := \operatorname{diag}[w_i^{-1}]_{i=1}^N$. Hence, we are first of all interested in solving the system

$$\Phi D \Phi^* \theta = y \tag{154}$$

and then we compute $\hat{x} = D\Phi^*\theta$. Notice that the system (154) has the general form

$$TT^*\theta = y, \tag{155}$$

with $T := \Phi D^{\frac{1}{2}}$. We consider then the application of CG to this system for the matrix $A = TT^*$. This approach leads to the modified conjugate gradient (MCG) method, presented in the appendix in Algorithm 9, proposed by J.T. King in [45]. It provides a sequence $(\theta^i)_{i \in \mathbb{N}}$ such that $\theta^i \in U_i := \operatorname{span}\{y, TT^*y, \ldots, (TT^*)^{i-1}y\}$, the Krylov subspace associated to (155), with the property that $\bar{x}^i := T^*\theta^i$ also minimizes $\left\| \bar{x}^i - \bar{x} \right\|_{\ell_2}$, where $\bar{x} = \underset{x \in \mathcal{F}_T(y)}{\arg\min} \|x\|_{\ell_2}$. Eventually, we compute $\hat{x} = D^{\frac{1}{2}}\bar{x}$.

Algorithm 9 Modified conjugate gradient (MCG) method

Input: initial vector $\theta^0 \in \mathbb{C}^m$, matrix $T \in \mathbb{C}^{m\times N}$, given vector $y \in \mathbb{C}^m$ and optionally a desired accuracy δ.

Set $\rho^0 = p^0 = y$ and $i = 0$
while $\rho^i \neq 0$ (or $\|\rho^i\|_{\ell_2} > \delta$) **do**
  $\alpha_i = \langle \rho^i, p^i\rangle_{\ell_2} / \|T^* p^i\|_{\ell_2}^2$
  $\theta^{i+1} = \theta^i + \alpha_i p^i$
  $\rho^{i+1} = y - TT^*\theta^{i+1}$
  $\beta_{i+1} = \langle T^* p^i, T^*\rho^{i+1}\rangle_{\ell_2} / \|T^* p^i\|_{\ell_2}^2$
  $p^{i+1} = \rho^{i+1} - \beta_{i+1} p^i$
  $i = i+1$
end while
Set $\bar{x}^{i+1} = T^*\theta^{i+1}$

The following theorem provides a precise rate of convergence of MCG. Additionally, we emphasize the monotonic decrease of the error $\|\hat{x}^i - \hat{x}\|_{\ell_2(w)}$.

Theorem 22. *Suppose that the matrix T be surjective. Then the sequence $(\bar{x}^i)_{i\in\mathbb{N}}$ generated by the Algorithm MCG converges to $\bar{x} = T^*(TT^*)^{-1}y$ in at most N steps, and*

$$\|\bar{x}^i - \bar{x}\|_{\ell_2} \leq \frac{2c_{TT^*}^i}{1 + c_{TT^*}^{2i}} \|\bar{x}^0 - \bar{x}\|_{\ell_2}, \tag{156}$$

for all $i \geq 0$, where c_{TT^*} is defined as in Theorem 21, and $\bar{x}^0 = T^*\theta^0$ is the initial vector. Moreover, by setting $D := \operatorname{diag}[w_i^{-1}]_{i=1}^N$, and $\hat{x}^i = D^{\frac{1}{2}}\bar{x}^i$ as well as $\hat{x} = D^{\frac{1}{2}}\bar{x}$, we obtain

$$\|\hat{x}^i - \hat{x}\|_{\ell_2(w)} \leq \frac{2c_{TT^*}^i}{1 + c_{TT^*}^{2i}} \|\hat{x}^0 - \hat{x}\|_{\ell_2(w)}. \tag{157}$$

Proof. By Theorem 21, we have

$$\left\|(TT^*)^{\frac{1}{2}}(\theta^i - \theta)\right\|_{\ell_2} \leq \frac{2c_{TT^*}^i}{1 + c_{TT^*}^{2i}} \left\|(TT^*)^{\frac{1}{2}}(\theta^0 - \theta)\right\|_{\ell_2},$$

for θ as given in (155). By the identity

$$\begin{aligned}
\left\|(TT^*)^{\frac{1}{2}}(\theta^i - \theta)\right\|_{\ell_2}^2 &= \langle (TT^*)^{\frac{1}{2}}(\theta^i - \theta), (TT^*)^{\frac{1}{2}}(\theta^i - \theta)\rangle_{\ell_2} \\
&= \langle (TT^*)(\theta^i - \theta), \theta^i - \theta\rangle_{\ell_2} \\
&= \langle T^*(\theta^i - \theta), T^*(\theta^i - \theta)\rangle_{\ell_2} \\
&= \langle \bar{x}^i - \bar{x}, \bar{x}^i - \bar{x}\rangle_{\ell_2} \\
&= \|\bar{x}^i - \bar{x}\|_{\ell_2}^2,
\end{aligned}$$

we obtain the assertion (156). □

References

1. B. Alexeev and R. Ward, "On the complexity of Mumford-Shah-type regularization, viewed as a relaxed sparsity constraint," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2787–2789, 2010.
2. M. Artina, M. Fornasier, and S. Peter, "Damping Noise-Folding and Enhanced Support Recovery in Compressed Sensing - Extended Technical Report," *ArXiv e-prints*, Nov. 2014.
3. M. Artina, M. Fornasier, and F. Solombrino, "Linearly constrained nonsmooth and nonconvex minimization," *SIAM J. Opt.*, vol. 23,, no. 3, pp. 1904–1937, 2012.
4. R. G. Baraniuk, M. Davenport, R. A. DeVore, and M. Wakin, *A simple proof of the restricted isometry property for random matrices*, Constr. Approx. 28 (2008), pp. 253–263.
5. A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci. **2** (2009), no. 1, 183–202.
6. T. Blumensath and M. E. Davies, *Iterative thresholding for sparse approximations*, J. Fourier Anal. Appl. **14** (2008), pp. 629–654.
7. T. Blumensath and M. E. Davies, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal., **27** (2009), no. 3, 265–274.
8. K. Bredies and D. A. Lorenz. Minimization of non-smooth, non-convex functionals by iterative thresholding. *Journal of Optimization Theory and Applications*, 2014. to appear.
9. E. J. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Ann. Statist.*, 37(5A):2145–2177, 10 2009.
10. E. J. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math. 59 (2006), pp. 1207–1223.
11. E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.
12. E. Arias-Castro and Y. C. Eldar, "Noise folding in compressed sensing," *IEEE Signal Process. Lett.*, pp. 478–481, 2011.
13. R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *Signal Processing Letters, IEEE*, 14(10):707–710, Oct 2007.
14. R. Chartrand and V. Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24(3):035020, 14, 2008.
15. R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3869–3872, March 2008.
16. S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by Basis Pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1999.
17. A. K. Cline, *Rate of convergence of Lawson's algorithm*, Math. Comp. **26** (1972), 167–176.
18. P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul. **4** (2005), 1168–1200.
19. S. Dahlke, M. Fornasier, and T. Raasch, *Multilevel preconditioning for adaptive sparse optimization*, Preprint 25, DFG-SPP 1324 Preprint Series, 2009
20. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, 1992.
21. I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math. 57 (2004), pp. 1413–1457.
22. I. Daubechies, R. A. DeVore, M. Fornasier, and C. S. Güntürk, *Iteratively re-weighted least squares minimization for sparse recovery*, Comm. Pure Appl. Math. **63** (2010), no. 1, 1–38.
23. M. Davenport, "The RIP and the NSP. @Connexions," http://cnx.org/content/m37176/1.5/, Apr. 2011.
24. D. L. Donoho and Y. Tsaig, *Fast solution of l1-norm minimization problems when the solution may be sparse*, IEEE Trans. Inform. Theory **54** (2008), 4789–4812.
25. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, *Least angle regression*, Ann. Statist. **32** (2004), 407–499.
26. I. Ekeland and R. Témam, *Convex analysis and variational problems*, SIAM, 1999.
27. H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Springer-Verlag, 1996.
28. M. Figueiredo and R. D. Nowak, *An EM algorithm for wavelet-based image restoration.*, IEEE Trans. Image Proc. **12** (2003), 906–916.
29. M. Figueiredo, R. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE J. Selected Topics in Signal Process. **4** (2007), no. 1, 586–597.
30. M. Fornasier, S. Peter, H. Rauhut, S. Worm, *Conjugate Gradient acceleration of iteratively re-weighted least squares methods*, preprint, pp. 47, 2005.
31. M. Fornasier. *Numerical Methods for Sparse Recovery*, pages 93–200. Radon Series on Computational and Applied Mathematics. De Gruyter, 2010.

32. M. Fornasier and R. Ward, "Iterative thresholding meets free-discontinuity problems," *Found. Comput. Math.*, vol. 10, no. 5, pp. 527–567, 2010.
33. S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing.* New York, NY: Birkhäuser/Springer, 2013.
34. I. F. Gorodnitsky and B. D. Rao, *Sparse signal reconstruction from limited data using FOCUSS: a recursive weighted norm minimization algorithm*, IEEE Transactions on Signal Processing **45** (1997), 600–616.
35. M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/∼boyd/graph_dcp.html.
36. J. Haupt, R. Baraniuk, R. Castro, and R. Nowak, "Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements," in *Proceedings of the 43rd Asilomar conference on Signals, systems and computers*, ser. Asilomar'09. Piscataway, NJ, USA: IEEE Press, 2009.
37. ——, "Sequentially designed compressed sensing," in *Proc. IEEE/SP Workshop on Statistical Signal Processing*, 2012.
38. J. Haupt, R. Castro, and R. Nowak, "Distilled sensing: Adaptive sampling for sparse detection and estimation," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6222–6235, 2011.
39. M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, Dec. 1952.
40. J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms I*, Vol. 305 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag: Berlin, 1996
41. P. W. Hollanda and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827, 1977.
42. K. Ito and K. Kunisch. A variational approach to sparsity optimization based on Lagrange multiplier theory. *Inverse Problems*, 30(1):015001, 23, 2014.
43. D. A. H. Jacobs. A generalization of the conjugate-gradient method to solve complex systems. *IMA journal of numerical analysis*, 6(4):447–452, 1986.
44. S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *A method for large-scale ℓ_1-regularized least squares problems with applications in signal processing and statistics*, IEEE Journal Sel. Top. Signal Process., **1** (2007), 606–617.
45. J. T. King. A minimal error conjugate gradient method for ill-posed problems. *Journal of Optimization Theory and Applications*, 60:297–304, 1989.
46. M.-J. Lai, Y. Xu, and W. Yin. Improved iteratively reweighted least squares for unconstrained smoothed ℓ_q minimization. *SIAM Journal on Numerical Analysis*, 51(2):927–257, 2013.
47. L. Landweber. An iterative formula for Fredholm integrals of the first kind. *American Journal of Mathematics*, 73:615-?624, 1951.
48. C. L. Lawson, *Contributions to the Theory of Linear Least Maximum Approximation*, 1961, Ph.D. thesis, University of California, Los Angeles.
49. I. Loris, *On the performance of algorithms for the minimization of* 1*-penalized functionals*, Inverse Problems **25** (2009), 035008.
50. S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, 2009.
51. S. G. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries.*, IEEE Trans. Signal Process. 41 (1993), pp. 3397–3415.
52. B. K. Natarajan, *Sparse approximate solutions to linear systems.*, SIAM J. Comput. 24 (1995), pp. 227–234.
53. V. Naumova, S. Peter, *Minimization of multi-penalty functionals by alternating iterative thresholding and optimal parameter choices*, Inverse Problems, vol. 30, 125003, 2014, 34 pp.
54. D. Needell, "Noisy signal recovery via iterative reweighted l1-minimization," in *Proceedings of the 43rd Asilomar conference on Signals, systems and computers*, ser. Asilomar'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 113–117. [Online]. Available: `http://dl.acm.org/citation.cfm?id=1843565.1843592`
55. Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM Studies in Applied Mathematics, vol. 13, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.
56. J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. New York: Springer, 2006.
57. P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock. An iteratively reweighted algorithm for non-smooth non-convex optimization in computer vision. *Technical Report*, 2014.
58. Z. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings*, Bull. Amer. Math. Soc. **73** (1967), 591–597.
59. M. Osborne, B. Presnell, and B. Turlach, *A new approach to variable selection in least squares problems*, IMA J. Numer. Anal. 20 (2000), pp. 389–403.

60. M. R. Osborne, *Finite algorithms in optimization and data analysis*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, John Wiley & Sons Ltd., Chichester, 1985.
61. M. Osborne, B. Presnell, and B. Turlach, *On the LASSO and its dual*, J. Comput. Graph. Statist. **9** (2000), 319–337.
62. A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in Applied Mathematics Series. Springer-Verlag GmbH, 2000.
63. R. Ramlau, G. Teschke, and M. Zhariy, *A compressive Landweber iteration for solving ill-posed inverse problems*, Inverse Problems **24** (2008), no. 6, 065013.
64. R. Ramlau and C. A. Zarzer. On the minimization of a Tikhonov functional with a non-convex sparsity constraint. *Electron. Trans. Numer. Anal.*, 39:476–507, 2012.
65. H. Rauhut, *Compressive Sensing and Structured Random Matrices*. In Theoretical Foundations and Numerical Methods for Sparse Recovery, Radon Series Comp. Appl. Math. deGruyter, 2010.
66. I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," http://cvxr.com/cvx, 2012.
67. J.-L. Starck, E. J. Candès, and D. L. Donoho, *Astronomical image representation by curvelet transform*, Astronomy and Astrophysics **298** (2003), 785–800.
68. J.-L. Starck, M. K. Nguyen, and F. Murtagh, *Wavelets and curvelets for image deconvolution: a combined approach*, Signal Proc. **83** (2003), 2279–2283.
69. R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B **58** (1996), 267–288.
70. J. Treichler, M. A. Davenport, and R. G. Baraniuk, "Application of compressive sensing to the design of wideband signal acquisition receivers," in *6th U.S. / Australia Joint Workshop on Defense Applications of Signal Processing (DASP)*, Lihue, Hawaii, Sept. 2009.
71. J. A. Tropp and D. Needell, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Appl. Comput. Harmon. Anal. **26** (2008), no. 3, 301–321.
72. J. A. Tropp, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. Inform. Theory **50** (2004), 2231–2242.
73. R. Vershynin, *Introduction to the non-asymptotic analysis of random matrices*, Compressed sensing, 210–268, Cambridge Univ. Press, Cambridge, 2012.
L. Vese, *A study in the BV space of a denoising-deblurring variational problem.*, Appl. Math. Optim. **44** (2001), 131–161.
74. C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Process.*, 7(6):813–824, 1998.
75. S. Voronin. *Regularization of Linear Systems with Sparsity Constraints with Applications to Large Scale Inverse Problems*. PhD thesis, Applied and Computational Mathematics Department, Princeton University, 2012.
76. C. A. Zarzer. On Tikhonov regularization with non-convex sparsity constraints. *Inverse Problems*, 25(2):025006, 13, 2009.