Available online at www.sciencedirect.com
ScienceDirect
Appl. Comput. Harmon. Anal. 23 (2007) 346–367
Applied and Computational Harmonic Analysis
www.elsevier.com/locate/acha

Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization

Michael Elad [a,*], Boaz Matalon [b], Michael Zibulevsky [b]

[a] *Department of Computer Science, Technion, Haifa 32000, Israel*
[b] *Department of Electrical Engineering, Technion, Haifa 32000, Israel*

Received 30 April 2006; revised 22 February 2007; accepted 22 February 2007

Available online 3 March 2007

Communicated by Richard Baraniuk

Abstract

This work addresses the problem of regularized linear least squares (RLS) with non-quadratic separable regularization. Despite being frequently deployed in many applications, the RLS problem is often hard to solve using standard iterative methods. In a recent work [M. Elad, Why simple shrinkage is still relevant for redundant representations? IEEE Trans. Inform. Theory 52 (12) (2006) 5559–5569], a new iterative method called parallel coordinate descent (PCD) was devised. We provide herein a convergence analysis of the PCD algorithm, and also introduce a form of the regularization function, which permits analytical solution to the coordinate optimization. Several other recent works [I. Daubechies, M. Defrise, C. De-Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math. LVII (2004) 1413–1457; M.A. Figueiredo, R.D. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Trans. Image Process. 12 (8) (2003) 906–916; M.A. Figueiredo, R.D. Nowak, A bound optimization approach to wavelet-based image deconvolution, in: IEEE International Conference on Image Processing, 2005], which considered the deblurring problem in a Bayesian methodology, also obtained element-wise optimization algorithms. We show that the last three methods are essentially equivalent, and the unified method is termed separable surrogate functionals (SSF). We also provide a convergence analysis for SSF. To further accelerate PCD and SSF, we merge them into a recently developed sequential subspace optimization technique (SESOP), with almost no additional complexity. A thorough numerical comparison of the denoising application is presented, using the basis pursuit denoising (BPDN) objective function, which leads all of the above algorithms to an iterated shrinkage format. Both with synthetic data and with real images, the advantage of the combined PCD-SESOP method is demonstrated.

Keywords: Least squares; Regularization; Inverse problems; Denoising; Shrinkage; Basis pursuit; Sparsity; Coordinate-descent; Proximal-point

* Corresponding author.
E-mail address: elad@cs.technion.ac.il (M. Elad).

1063-5203/$ – see front matter
doi:10.1016/j.acha.2007.02.002

1. Introduction

In this work we focus on the problem of linear least squares with non-quadratic regularization, or regularized least squares (RLS) in short, i.e. the minimization of

$$f(\mathbf{z}) = \|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2 + \rho(\mathbf{z}), \tag{1.1}$$

where $\rho(\mathbf{z}) = \sum_j \rho_j(\mathbf{z}(j))$ and $\|\cdot\|$ stands for the l_2 norm throughout this paper. The first term weighs the difference between $\mathbf{x} = \mathbf{A}\mathbf{z}$ and the noisy observation $\mathbf{b} = \mathbf{x} + \mathbf{n}$, where $\mathbf{x}, \mathbf{b}, \mathbf{n}$ are of length N, $\mathbf{z}$ is K long and $\mathbf{A}$ is a full rank matrix of size $N \times K$. Without the second term, if $N \leqslant K$, the solution of (1.1) would involve the so-called *pseudo-inverse*, i.e. $\mathbf{z}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. In case $N < K$ there are infinitely many solutions and the above pseudo-inverse formula becomes irrelevant. The second term, which is a sum of element-wise penalties, serves as a regularizer.

The above formulation is a general one, adequate for numerous applications, such as: (1) denoising [10], where the objective function is given by

$$f(\mathbf{z}) = \|\Phi\mathbf{z} - \mathbf{b}\|^2 + \rho(\mathbf{z}). \tag{1.2}$$

Φ is the so-called *dictionary*, and its columns are known as *atoms*. It is assumed that the desired signal is a linear combination of these atoms, i.e. $\mathbf{x} = \Phi\mathbf{z}$. Equation (1.2) is the well-known basis-pursuit denoising (BPDN) problem [5]. However, here we allow for a general regularizer ρ, instead of just $\rho_j(z) \propto |z|$, which was used in [5]. Section 5 provides a thorough comparison of the discussed algorithms with the BPDN problem; (2) deblurring [7], in which $\mathbf{A} = \mathbf{K}\Phi$, $\mathbf{K}$ represents a blur operator; and (3) tomography [20], which is similar to deblurring, with $\mathbf{K}$ representing the Radon transform.

Another way of obtaining the RLS problem (1.1) is using the maximum-a-posteriori-probability (MAP) estimator, i.e.

$$\hat{\mathbf{z}} = \arg\max_{\mathbf{z}} \big\{\log p(\mathbf{b}|\mathbf{z}) + \log p(\mathbf{z})\big\}. \tag{1.3}$$

This formulation develops into (1.1), if we assume that $\mathbf{n}$ is a white-Gaussian noise and that $\mathbf{z}$ is distributed according to $p(\mathbf{z}(j)) \propto \exp(-\text{Const} \cdot \rho_j(\mathbf{z}(j)))$. The function ρ depends on the characterization of the specific signals at hand. One common choice of ρ involves the l_p norm, i.e. $\rho(\mathbf{z}) = \|\mathbf{z}\|_p^p$, $p \in [0, \infty]$. Within this family, $p = 1$ is the closest to the ideal sparsity-inducing prior (i.e. $p = 0$), still yielding a convex objective function. In this paper we will also consider a general penalty function ρ.

It is well known (see, for example, [10,19,27]) that for a unitary transform, i.e. $\mathbf{A}^{-1} = \mathbf{A}^T$, there is an explicit solution for (1.1) called *shrinkage*, which amounts to a simple element-wise operation (see Section 2.1 for details). In the case where $\rho(\mathbf{z}) = \|\mathbf{z}\|_p^p$ with $p \leqslant 2$, the corresponding non-linear operation shrinks the absolute value of each coefficient, hence the name of this solution. This simple scheme relies heavily both on the white-Gaussianity assumption, as well as on $\mathbf{A}$ being unitary. Many redundant transforms were developed in recent years (e.g., [9,22,31, 32,35]), especially for images, enabling more parsimonious representations, which orthogonal transforms find difficult to achieve. Although not being the solution of (1.1) any longer, shrinkage is still extensively used for non-unitary and even redundant transforms, yielding much better results than using orthogonal transforms.

A recent work by Elad [11], which devised an iterative algorithm for the RLS problem, herein termed parallel coordinate descent (PCD), managed to explain this behavior. It is motivated by the coordinate-descent method (see, for example, [25]), in which the objective function is minimized one coordinate at a time. PCD suggests collecting the various entry-wise updates into a single vector, followed by a line-search along this direction. The resulting algorithm involves shrinkage in each iteration, consisting of multiplication by $\mathbf{A}^T$ (instead of $\mathbf{A}^{-1}$) and $\mathbf{A}$.

Several recent works have obtained similar algorithms by solving the deblurring problem, i.e. reconstructing signals corrupted by blur and noise. Interestingly, each had a different motivation: [13] employed the notion of missing data, naturally rendering the expectation-maximization (EM) technique; [7] and [14] used a sequence of surrogate functionals that are minimized via shrinkage. One should wonder about the connections between all of these methods. In this work we show that by posing the objective as in (1.1), these three methods are essentially equivalent, other than a few technical differences.

This paper is organized as follows: In Section 2 we provide a convergence proof of the PCD algorithm [11], and analyze its asymptotic convergence rate. We also propose a form of $\rho_j(z)$, which both approximates the l_p norm for $p \in (1, 2)$, and yields an explicit PCD step. Section 3 discusses the other three recently developed algorithms, shows

their equivalence, and provides an alternative convergence proof to that proposed in [7]. In addition, the asymptotic behavior is analyzed, and a comparison with PCD is made. In Section 4 we suggest a way of accelerating all of these algorithms, by recruiting a recent optimization technique called SESOP [28]. Section 5 provides a thorough numerical comparison of the discussed algorithms and of other classic optimization algorithms, while referring to the theoretical asymptotic analysis, demonstrating the advantage of the combined PCD-SESOP method.

2. Parallel coordinate descent (PCD)

2.1. Formulation

This subsection expresses ideas of [11] and is presented here for completeness, while the following subsections are completely new. Assume that within an iterative optimization process of minimizing (1.1), we have the kth estimate $\mathbf{z}_k$. To calculate the next estimate, $\mathbf{z}_{k+1}$, the jth entry is updated, assuming all other entries are fixed, i.e.

$$\mathbf{z}_{k+1}(j) = \arg\min_z \left\| \left[\mathbf{A}\mathbf{z}_k - \mathbf{a}_j \mathbf{z}_k(j)\right] + \mathbf{a}_j z - \mathbf{b} \right\|^2 + \rho_j(z), \tag{2.1}$$

where $\mathbf{a}_j$ denotes the jth column of $\mathbf{A}$. Such an algorithm, which makes a descent along a different coordinate at a time, either cyclically or according to some other rule, is often referenced to as the coordinate-descent algorithm [25].

As [11] shows, the choice of $\rho_j(z) = \mathbf{w}(j)|z|$, where $\mathbf{w} = [\mathbf{w}(1), \ldots, \mathbf{w}(K)]^T$ denotes a positive weights vector, leads to an analytic updating rule, essentially a simple one-dimensional soft-shrinkage operation. However, in most cases this algorithm is not practical, since we need to extract the jth column of $\mathbf{A}$ (the jth dictionary atom) for the jth coordinate updating. Many transforms, e.g. the wavelet transform, are computed via a fast recursive scheme, rather than an actual matrix–vector multiplication. Thus, the use of isolated atoms is computationally inefficient, ruling out employing the coordinate-descent algorithm.

To overcome this difficulty, the *parallel coordinate descent* (PCD) algorithm is proposed. Rather than updating each coordinate sequentially, the whole set of coordinates is updated simultaneously at the current point $\mathbf{z}_k$. Of course, such an approach does not guarantee anymore decrease of the objective function. Nevertheless, since a non-negative linear combination of descent directions is also a descent one, a line-search along that direction does ensure descending. Specifically, define the parallel coordinate descent algorithm for minimizing $f(\mathbf{z})$ as

(1) compute the components of the descent direction by

$$\mathbf{d}_k(j) = \arg\min_\alpha f(\mathbf{z}_k + \alpha \mathbf{e}_j), \quad j = 1, \ldots, K, \tag{2.2}$$

where $\mathbf{e}_j$ stands for the jth elementary unit vector;

(2) perform line-search along $\mathbf{d}_k$, i.e. $\mathbf{z}_{k+1} = \mathbf{z}_k + \mu_k \mathbf{d}_k$.

Since the concept of optimization per element will be reiterated in the next section, it will be termed as element-wise optimization.

In the case where $\rho_j(z) = \mathbf{w}(j)|z|$, the direction $\mathbf{d}_k$ can be explicitly expressed. Denote $\operatorname{diag}\{\mathbf{A}^T\mathbf{A}\}$ as the diagonal matrix, containing the atoms' norms $\{\|\mathbf{a}_j\|^2\}$ in its diagonal. We also define $\hat{\mathbf{w}} = \frac{1}{2}\operatorname{diag}^{-1}\{\mathbf{A}^T\mathbf{A}\}\mathbf{w}$, and

$$\mathbf{v}(\mathbf{A}, \mathbf{b}, \mathbf{z}_k) = \mathbf{z}_k + \operatorname{diag}^{-1}\left\{\mathbf{A}^T\mathbf{A}\right\}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{z}_k). \tag{2.3}$$

Then $\mathbf{d}_k$ is given by

$$\mathbf{d}_k = \mathcal{S}_{\hat{\mathbf{w}}}\left\{\mathbf{v}(\mathbf{A}, \mathbf{b}, \mathbf{z}_k)\right\} - \mathbf{z}_k, \tag{2.4}$$

where

$$\mathcal{S}_\delta\{z\} = \operatorname{sign}(z)\big(|z| - \delta\big)_+ \tag{2.5}$$

is the well-known soft-shrinkage function [10], operating entry-wise upon a vector

$$\mathcal{S}_{\mathbf{x}}\{\mathbf{y}\} = \left[\mathcal{S}_{\mathbf{x}(1)}\left\{\mathbf{y}(1)\right\}, \ldots, \mathcal{S}_{\mathbf{x}(K)}\left\{\mathbf{y}(K)\right\}\right]^T. \tag{2.6}$$

From (2.1) we can also easily obtain a formula similar to (2.4) for $\rho_j(z) = \mathbf{w}(j)|z|^2$

$$\mathbf{d}_k = \left(\operatorname{diag}\left\{\mathbf{A}^T\mathbf{A}\right\} + 2\mathbf{W}\right)^{-1}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{z}_k), \tag{2.7}$$

where $\mathbf{W}$ is a diagonal matrix, with $\mathbf{w}$ in its diagonal. This rule is in fact a Landweber step [21,36], except for the normalization by $\{\|\mathbf{a}_j\|^2 + \mathbf{w}(j)\}$, and the inevitable line-search along $\mathbf{d}_k$.

As is clearly evident, the above formulae require applying the synthesis operator $\mathbf{A}$ once, and its adjoint $\mathbf{A}^T$ once. There is no need to extract the dictionary atoms in any way, and the calculation of the atoms' norms can be made off-line. Estimating these norms can be achieved as follows: observe that a white Gaussian vector $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I})$ satisfies $\operatorname{Cov}\{\mathbf{A}^T\mathbf{v}\} = \mathbf{A}^T E\{\mathbf{v}\mathbf{v}^T\}\mathbf{A} = \mathbf{A}^T\mathbf{A}$. Hence, we can apply $\mathbf{A}^T$ on a few realizations of a pseudo-random vector, and obtain the atoms' norms directly by averaging the corresponding coefficients' squared values. A simpler and faster computation may be found for structured transforms, such as the wavelet transform (see [19] for example).

2.2. Smoothing functions

In contrast to (2.4) and (2.7), an analytical solution to (2.1) with $\rho_j(z) = \mathbf{w}(j)|z|^p$ does not exist for $1 < p < 2$, since the consequent optimality equations cannot be explicitly solved [27]. A possible alternative is to use a smooth and convex approximation of the prior ρ, a one that enables an analytical solution. Also, the objective function should be smoothed anyway for the convergence study that follows. We have chosen the following family of smoothed objective functions for $\rho_j(z) = \mathbf{w}(j)|z|^p$, with $1 < p < 2$:

$$f_s(\mathbf{z}) = \|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2 + \mathbf{w}^T\varphi_s(\mathbf{z}) = \|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2 + \sum_j \mathbf{w}(j)\varphi_s\big(\mathbf{z}(j)\big), \quad s \in (0, \infty). \tag{2.8}$$

We define the one-dimensional function φ_s as

$$\varphi_s(z) = |z| - s\ln\big(1 + |z|/s\big), \quad s \in (0, \infty). \tag{2.9}$$

Its first and second derivatives are

$$\varphi_s'(z) = \frac{|z|\operatorname{sign}(z)}{s + |z|}, \qquad \varphi_s''(z) = \frac{s}{(s + |z|)^2} > 0, \tag{2.10}$$

showing that the function is twice differentiable and strictly convex. We note here that similar smoothing functions, which enable an explicit update rule, exist for $p < 1$ as well.

Let us examine the function for different values of s. For $s \gg 1$,

$$\varphi_s(z) = |z| - s\ln\big(1 + |z|/s\big) \simeq |z| - s\big[\big(|z|/s\big) - \big(|z|/s\big)^2\big] = z^2/s. \tag{2.11}$$

For $s \ll 1$,

$$\varphi_s(z) = |z| - s\ln\big(1 + |z|/s\big) \simeq |z| - s\ln\big(|z|/s\big) \simeq |z|. \tag{2.12}$$

As a result, $\varphi_s(z)$ with a small s may be used as a smooth approximation of $|z|$, a conclusion which will be utilized later on. It can be graphically verified (see Fig. 1) that for each $p \in (1, 2)$, there is an $s \in (0, \infty)$, such that φ_s approximates $|z|^p$ within $z \in [-1, 1]$, up to a manually-calibrated scaling factor which depends on s.

To obtain a shrinkage-like updating rule for the parallel coordinate descent algorithm, we plug φ_s into (2.1) and equate its derivative to 0, yielding

$$0 = 2\mathbf{a}_j^T\left(\left[\mathbf{A}\mathbf{z}_k - \mathbf{a}_j\mathbf{z}_k(j)\right] + \mathbf{a}_j z - \mathbf{b}\right) + \mathbf{w}(j)\frac{|z|\operatorname{sign}(z)}{s + |z|}. \tag{2.13}$$

This is a quadratic equation for either $z \geqslant 0$ or $z \leqslant 0$, thus an analytic solution can be obtained. The subsequent parallel coordinate descent iteration has the same form as (2.4), only that the soft-shrinkage function $\mathcal{S}_\delta$ (2.5) is replaced with

$$\mathcal{S}_{\delta,s}(z) = \begin{cases} \frac{1}{2}\big(z - \delta - s + \sqrt{(\delta + s - z)^2 + 4sz}\big), & z \geqslant 0, \\ -\mathcal{S}_{\delta,s}(-z), & z < 0. \end{cases} \tag{2.14}$$

Figure 2 illustrates several smoothed soft-shrinkage functions.

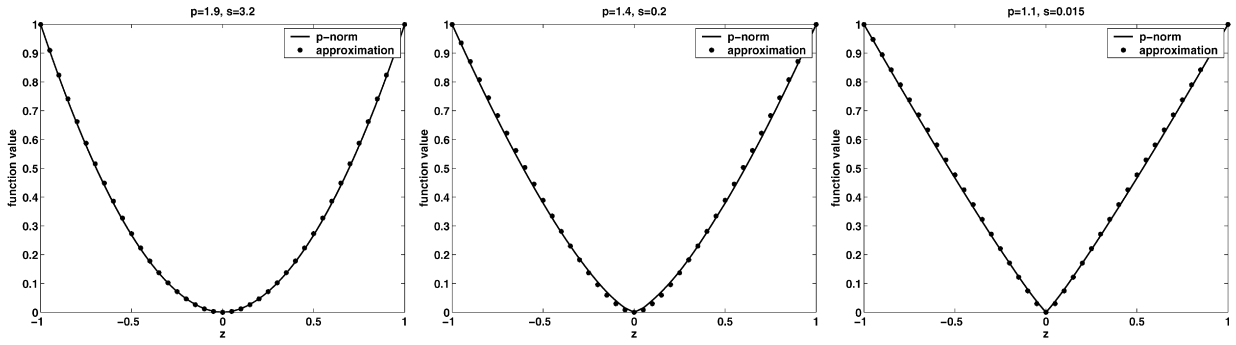

Fig. 1. Smoothing functions for several parameter values.

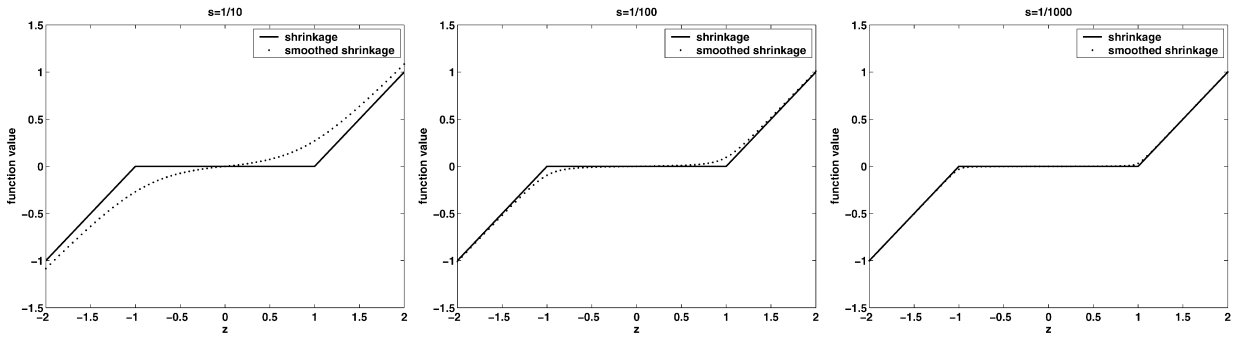

Fig. 2. Soft-shrinkage functions for several parameter values (with $\delta = 1$). From left to right: $s = 1/10, 1/100, 1/1000$.

2.3. Convergence proof

The question arising now is whether the parallel coordinate descent algorithm converges to a local minimum of (1.1). Since $\mathbf{d}_k$ is a descent direction, the objective function always decreases, yet this fact does not guarantee convergence. However, apparently the PCD algorithm does indeed converge under mild conditions, as we shall prove now.

Theorem 2.1. *Consider a function $f(\mathbf{z})$, satisfying*

(D0) *The level set $\mathcal{R} = \{\mathbf{z}:\ f(\mathbf{z}) \leqslant f(\mathbf{z}_0)\}$ is compact.*
(D1) *The Hessian $\mathbf{H}(\mathbf{z}) = [\partial^2 f/\partial z_i \partial z_j]_{i,j}$ is bounded in $\mathcal{R}$, i.e. $\|\mathbf{H}(\mathbf{z})\| \leqslant M, \forall \mathbf{z} \in \mathcal{R}$.*

Define the parallel coordinate descent algorithm as $\mathbf{z}_{k+1} = \mathbf{z}_k + \mu_k \mathbf{d}_k$, where

$$\mathbf{d}_k(j) = \arg\min_{\alpha} f(\mathbf{z}_k + \alpha \mathbf{e}_j), \quad j = 1, \ldots, K. \tag{2.15}$$

$\mathbf{e}_j$ stands for the jth elementary unit vector, and μ_k is obtained by a line-search along $\mathbf{d}_k$. Then any limit point $\bar{\mathbf{z}}$ of the sequence $\{\mathbf{z}_k\}$ is a stationary point of f, i.e. $\nabla f(\bar{\mathbf{z}}) = 0$, and $f(\mathbf{z}_k) \to f(\bar{\mathbf{z}})$.

Proof. Our proof will be conducted as follows: First, we will assume by contradiction that the gradient in a limit point is not zero. The gradient's norm is thus positive for large-enough indices of the sequence of solutions. Next, we will calculate the scalar-product between the gradient and the descent direction, and lower-bound its absolute value (again, for large-enough indices). It will be then shown that the difference in the objective function between consecutive iterations depends on this product, unavoidably yielding an infinite decrease in the function value. This will come as a clear negation with the function being bounded below in a compact set, allowing us to conclude that the gradient at the limit point is indeed zero. For a complete and detailed proof, see Appendix A. □

Now, consider a case where the initial point $\mathbf{z}_0$ lies in a compact level set, which has no stationary points other than a set of minimizers $\{\mathbf{z}^*\}$ (all having the same function value within the specified region). The theorem implies that $f(\mathbf{z}_k)$ converges to the minimum $f(\mathbf{z}^*)$. Moreover, in the case of a unique minimizer $\mathbf{z}^*$, it is straightforward to show that the series $\mathbf{z}_k$ converges to $\mathbf{z}^*$. Of course, if this minimizer is global, than the theorem yields global convergence.

Let us examine whether the theorem's assumptions are met by our specific f, defined in (1.1). The first term is quadratic in $\mathbf{z}$, with Hessian $\mathbf{H}(\mathbf{z}) = 2\mathbf{A}^T\mathbf{A} \succeq 0$. This term does not fulfill the first assumption, surely when $\mathbf{A}$ is redundant. Hence, for the theorem to hold, it is sufficient that $|\rho_j''|$ is bounded and that ρ_j is strictly unimodal, for every j. As an example, if $\rho_j(z) = \mathbf{w}(j)|z|^p$ (with $\mathbf{w} > 0$), both of these conditions are satisfied for $p > 1$. When $p \leqslant 1$, however, $\rho_j(z)$ is not differentiable at $z = 0$, and so the proof is not valid. To nevertheless use the theorem, the l_p norm may be smoothed at $z = 0$.

As it happens, we have supplied earlier a smoothed version for the l_1 norm (see Section 2.2). Therefore, the function f_s, defined in (2.8), does indeed satisfy the theorem's assumptions as required. In addition, it is strictly convex, and so the global minimum of (2.8) is achieved. As previously stated, f_s with $s \ll 1$ is a good approximation of f with the prevalent choice of $\rho_j(z) = \mathbf{w}(j)|z|$. As a result, minimizing f_s with a small smoothing parameter can be used to solve the original problem. In practice, as later simulations will show, the optimization process becomes less efficient as s is reduced. However, increasing s might damage the solution's accuracy relative to the exact l_1 norm case. Therefore, the chosen value of s is a compromise between accuracy and speed.

2.4. Asymptotic convergence rate

Now that we have established the convergence of the PCD algorithm, it is worthwhile exploring the rate of convergence. In some cases, e.g. denoising, where a good-enough solution is often reached within a small number of iterations, only the first few iterations are important and should be studied. Yet, such complexity estimate is often hard to obtain, and thus the asymptotic one is analyzed instead, as we will do here. Within the scope of this work we will concentrate solely on the quadratic case, projecting to the general case by means of the second-order Taylor expansion at $\mathbf{z}^*$. A concise analysis of the general non-quadratic function is left to future work.

Consider a strictly convex quadratic function $f(\mathbf{z}) = \mathbf{z}^T\mathbf{Q}\mathbf{z}$ with $\mathbf{Q} \succ 0$. As we did in developing PCD, suppose we have the kth solution $\mathbf{z}_k$, and we want to update only the jth entry, i.e.

$$\mathbf{z}_{k+1}(j) = \arg\min_z \mathbf{q}_{jj}\big(z - \mathbf{z}_k(j)\big)^2 + 2\mathbf{q}_j^T\mathbf{z}_k\big(z - \mathbf{z}_k(j)\big) + \mathbf{z}_k^T\mathbf{Q}\mathbf{z}_k. \tag{2.16}$$

$\mathbf{q}_{jj}$ denotes the jth diagonal element, while $\mathbf{q}_j$ is the jth column of $\mathbf{Q}$. This trivially yields

$$\mathbf{z}_{k+1}(j) = \mathbf{z}_k(j) - \frac{\mathbf{q}_j^T\mathbf{z}_k}{\mathbf{Q}_{jj}}, \quad \forall j. \tag{2.17}$$

As discussed earlier, the descent direction $\mathbf{d}_k$ is constructed by collecting the various $\{\mathbf{z}_{k+1}(j)\}$ into a vector and subtracting $\mathbf{z}_k$, so in this case

$$\mathbf{d}_k = -\mathrm{diag}^{-1}\{\mathbf{Q}\}\mathbf{Q}\mathbf{z}_k. \tag{2.18}$$

Since $\nabla f(\mathbf{z}) = 2\mathbf{Q}\mathbf{z}$ and $\mathbf{H} = 2\mathbf{Q}$, the PCD direction equals the steepest-gradient direction [15], preconditioned by the Hessian's diagonal, which is the well-known Jacobi-method. If we denote $\mathbf{\Lambda} = \mathrm{diag}^{-1}\{\mathbf{H}\}$, then we have $\mathbf{d}_k = -\mathbf{\Lambda}\nabla f(\mathbf{z}_k)$. This method has proven asymptotic rate formulae [30], specifically

$$f(\mathbf{z}_{k+1}) - f(\mathbf{z}^*) \leqslant \left(\frac{M-m}{M+m}\right)^2 \big(f(\mathbf{z}_k) - f(\mathbf{z}^*)\big), \tag{2.19}$$

where m and M indicate the smallest and largest eigen-values of $\mathbf{\Lambda}^{1/2}\mathbf{H}\mathbf{\Lambda}^{1/2}$, respectively.

Although the RLS objective function (1.1) is not quadratic (unless $\rho_j(z) \propto |z|^2$), it can be approximated by its second-order Taylor expansion at $\mathbf{z}^*$,

$$f(\mathbf{z}) \approx f(\mathbf{z}^*) + (\mathbf{z} - \mathbf{z}^*)^T\mathbf{H}(\mathbf{z}^*)(\mathbf{z} - \mathbf{z}^*). \tag{2.20}$$

In the case of $\rho_j(z) \propto |z|$, where this expansion does not necessarily exist, the smoothed version (2.8) must be employed instead, as will be done in our simulations. In Section 5, the above rate of convergence will be compared to the experimental one, measured once the algorithm reaches small-enough gradients' norms.

3. Separable surrogate functionals (SSF)

This section describes an alternative element-wise optimization algorithm for RLS, which is based on several recent methods designed for deblurring [7,13,14]. Two of these methods [7,14] make use of separable surrogate functionals (SSF), giving this section its name. The other method [13] is driven by the expectation-maximization (EM) technique [8], leading to a very similar algorithm. This section is constructed as follows: First, we will describe the SSF method for the RLS problem (1.1), unifying the different points of view exhibited by these recent works. Thereafter, each work will be briefly reviewed, showing that the subsequent algorithms are essentially the same, when applied to the RLS problem. Then, a global convergence proof will be presented, as well as an asymptotic behavior analysis. Finally, a short discussion and a comparison with the PCD algorithm will be provided.

3.1. Formulation

The following development results from adapting the works of Daubechies et al. [7] and of Figueiredo et al. [14], which will be described in turn, to the RLS problem (1.1). Both rely on the optimization transfer method [23], also referred to as the bound optimization approach [18]. Suppose we have a function $f(\mathbf{z})$, which is too difficult to minimize as is. The well-known EM algorithm [8] suggests an iterative scheme, in which the sequence of estimates $\mathbf{z}_k$ stems from

$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} Q(\mathbf{z}|\mathbf{z}_k). \tag{3.1}$$

Within the EM framework, the so-called Q-function is constructed by introducing some missing-data variables. This construction ensures the monotone decrease of the objective function $f(\mathbf{z})$.

Yet, for such a monotone behavior to hold, the following *key property* is sufficient

$$Q(\mathbf{z}|\mathbf{z}_k) \geqslant f(\mathbf{z}), \quad \forall \mathbf{z}, \tag{3.2}$$

with equality for $\mathbf{z} = \mathbf{z}_k$. Using the last property and (3.1), we can easily deduce that

$$f(\mathbf{z}_{k+1}) \leqslant Q(\mathbf{z}_{k+1}|\mathbf{z}_k) \leqslant Q(\mathbf{z}_k|\mathbf{z}_k) = f(\mathbf{z}_k). \tag{3.3}$$

Therefore, EM-like algorithms can be built by upper-bounding the objective function, without calling for missing-data considerations. This is exactly what the optimization transfer is all about, and we shall show its use for the RLS problem, similarly to [7] and [14].

The idea is to minimize a non-quadratic separable Q-function, also referred to in [7] as the *surrogate functional*. If we denote

$$\psi(\mathbf{z}) = \|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2, \tag{3.4}$$

then the desired Q-function is

$$Q(\mathbf{z}|\mathbf{z}_k) = \psi(\mathbf{z}_k) + \nabla\psi(\mathbf{z}_k)^T(\mathbf{z} - \mathbf{z}_k) + (\mathbf{z} - \mathbf{z}_k)^T\mathbf{\Lambda}(\mathbf{z} - \mathbf{z}_k) + \rho(\mathbf{z}), \tag{3.5}$$

where

$$\mathbf{\Lambda} \succeq \mathbf{A}^T\mathbf{A} \tag{3.6}$$

is some diagonal matrix. Notice that the quadratic term is decoupled, making the optimality equations arising from (3.1) decouple as well. This explains the name separable surrogate functionals (SSF), given to this section.

It can be easily seen that this Q-function satisfies (3.2), if we write the RLS problem (1.1) as

$$f(\mathbf{z}) = \psi(\mathbf{z}_k) + \nabla\psi(\mathbf{z}_k)^T(\mathbf{z} - \mathbf{z}_k) + (\mathbf{z} - \mathbf{z}_k)^T\mathbf{A}^T\mathbf{A}(\mathbf{z} - \mathbf{z}_k) + \rho(\mathbf{z}). \tag{3.7}$$

Therefore,

$$Q(\mathbf{z}|\mathbf{z}_k) = f(\mathbf{z}) + \phi(\mathbf{z}, \mathbf{z}_k), \tag{3.8}$$

where

$$\phi(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T\big(\mathbf{\Lambda} - \mathbf{A}^T\mathbf{A}\big)(\mathbf{u} - \mathbf{v}) \geqslant 0. \tag{3.9}$$

In addition, notice that the Q-function is indeed separable, reiterating the concept of element-wise optimization, introduced in the previous section.

The subsequent iterative update, for $\rho_j(z) = \mathbf{w}(j)|z|$, is (see [7] for details)

$$\mathbf{z}_{k+1} = \mathcal{S}_{\bar{\mathbf{w}}}\left\{\mathbf{z}_k + \mathbf{\Lambda}^{-1}\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{z}_k)\right\}, \tag{3.10}$$

where $\bar{\mathbf{w}} = \frac{1}{2}\mathbf{\Lambda}^{-1}\mathbf{w}$ and $\mathcal{S}_\delta$ is given by (2.5). To obtain explicit formulae for $p \in (1, 2)$ as well, we may use the smoothed objective functions (2.8). The above formula is strikingly similar to that obtained by the PCD algorithm (2.4). In fact, ignoring the line-search, substituting $\mathbf{\Lambda}$ by $\operatorname{diag}\{\mathbf{A}^T\mathbf{A}\}$ gives the same updating rule. In Section 3.4 we will discuss in detail the consequences of this last observation. Now, the functionals of (3.9) are actually quadratic and strictly convex in $(\mathbf{u} - \mathbf{v})$. Hence, adding $\phi(\mathbf{z}, \mathbf{z}_k)$ to $f(\mathbf{z})$ to obtain the Q-function can be interpreted as promoting proximity between subsequent estimates. This is a special case of the proximal-point algorithm [33], also referred to as prox-algorithm.

The above algorithm (3.1)–(3.6) is based on several recent papers, which developed similar algorithms in the context of linear inverse problems. In [7], the deblurring problem was formulated as the minimization of

$$f(\mathbf{x}) = \|\mathbf{K}\mathbf{x} - \mathbf{b}\|^2 + \mathbf{w}^T\|\mathbf{T}\mathbf{x}\|_p^p, \quad p \in [1, 2], \tag{3.11}$$

where $\mathbf{K}$ denotes the blur operator and $\mathbf{T}$ represents a unitary forward transform. The minimization was made by employing the following sequence of functions:

$$\phi(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T\left(\mathbf{\Lambda} - \mathbf{K}^T\mathbf{K}\right)(\mathbf{u} - \mathbf{v}). \tag{3.12}$$

We can obtain the RLS setting (1.1) simply by utilizing a general operator $\mathbf{A}$ instead of the blur $\mathbf{K}$, choosing $\mathbf{T} = \mathbf{I}$, denoting $\mathbf{z} = \mathbf{x}$ and allowing for a general prior ρ, thereby obtaining exactly the same algorithm. The work in [7] also proves global convergence of the iterative algorithm to the minimizer of (3.11). We will present in the next subsection an alternative proof, valid for any penalty ρ, based on the identification of this algorithm as a proximal-point algorithm.

In [14], Figueiredo and Nowak presented a slightly different setting, in which any tight frame $\mathbf{W}$ could be used (e.g., the translation-invariant wavelet transform [26]). It was posed as minimizing

$$f(\mathbf{z}) = \|\mathbf{K}\mathbf{W}\mathbf{z} - \mathbf{b}\|^2 + w\|\mathbf{z}\|_p^p, \tag{3.13}$$

where $\mathbf{W}$ satisfies $\mathbf{W}\mathbf{W}^T = \mathbf{I}$ (tightness with a constant of 1). The minimization was motivated by the bound-optimization approach, using the family

$$\phi(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T\left(\mathbf{I} - \mathbf{W}^T\mathbf{K}^T\mathbf{K}\mathbf{W}\right)(\mathbf{u} - \mathbf{v}), \tag{3.14}$$

which are all convex thanks to the assumptions $\|\mathbf{K}\| \leqslant 1$ and $\mathbf{W}\mathbf{W}^T = \mathbf{I}$. In this case, we obtain the RLS problem by generalizing the prior ρ, by denoting $\mathbf{A} = \mathbf{K}\mathbf{W}$ and by removing any constraints of the frame's tightness. This of course forces us to replace $\mathbf{I}$ in (3.14) with $\mathbf{\Lambda}$, satisfying $\mathbf{\Lambda} \succ \mathbf{A}^T\mathbf{A}$, thereby yielding the SSF once again. In the same work, it is also suggested deploying the bound-optimization approach for the prior $w\|\mathbf{z}\|_p^p$ for any $p \in (1, 2)$ and $p \in (0, 1)$, consequently giving a closed-form update rule, which normally cannot be attained (see [27]). Our work focuses on bounding the quadratic term alone, and using a smoothed version for the l_p norm.

An earlier work by Figueiredo and Nowak [13] handled the very same deblurring problem posed in (3.13), apart from only dealing with an orthonormal dictionary $\mathbf{W}$. By introducing the notion of missing data, the EM approach [8] was naturally utilized. Adapting their setting to the RLS problem, as was done in the previous paragraph, we once again get the element-wise optimization algorithm defined in (3.10), this time with $\mathbf{\Lambda} = \lambda_{\max}\{\mathbf{A}\mathbf{A}^T\}\mathbf{I}$. In Section 3.4 we will see that this is the most practical choice for $\mathbf{\Lambda}$ (only adding a small $\varepsilon > 0$), verifying our claimed equivalence of the three mentioned works, posed as a RLS problem.

A recent paper by Combettes and Wajs [6] managed to describe a large class of inverse problems as a minimization of the sum of two functions with certain properties. This work rigorously analyzed the convergence properties of a forward–backward algorithm, which employs proximity-operators, such as the one described above. The consequent algorithm for the RLS problem takes a generalized form of (3.10), allowing for, among others, an iteration-dependent diagonal matrix $\mathbf{\Lambda}_k$. It was shown that taking $\mathbf{\Lambda}_k \succ \frac{1}{2}\mathbf{A}\mathbf{A}^T$, rather than $\mathbf{\Lambda}_k \succ \mathbf{A}\mathbf{A}^T$, is sufficient to prove convergence. In that respect, their result is more powerful than the one presented herein.

We mention here another recent work by Bioucas-Dias [3], which deals with the deblurring problem (3.13), only that the prior ρ is restricted to the Gaussian-scale-mixture (GSM) family [1], containing the l_p norm for example.

$p(z)$ is a GSM distribution if $z = \sqrt{\alpha}u$, where α is a positive random variable, independent of $u \sim \mathcal{N}(0, 1)$. Like [13], the work in [3] also recruited the EM technique to minimize f, yet this time with the scale variables $\{\alpha_j\}$ playing the missing data role. Each iteration of the subsequent algorithm requires solving a huge linear system, which is approximated by employing a few iterations of a second-order process. This algorithm is termed in [3] generalized-EM (GEM), since the M-step does not perform exact maximization, only increments the Q-function. Although this method can be adapted to the RLS formulation (1.1), it is not an element-wise optimization algorithm, and is thus outside the scope of our work.

3.2. Convergence proof

As we have said, global convergence of the discussed algorithm was established in [7], for the l_p norm with $p \in [1, 2]$. Identifying the discussed method as a special case of the prox-algorithm allows us to bring an alternative proof for a general ρ. This proof is similar in spirit to the one available in [2], yet with different assumptions.

The following proof uses the notions of subgradients and subdifferentials [17], which we shall describe here for completeness. Consider a convex function g defined on a convex open set $\mathcal{R}$. Then a vector $\mathbf{u}$ is a subgradient of g at $\mathbf{z}_0$, if

$$g(\mathbf{z}) - g(\mathbf{z}_0) \geqslant \mathbf{u}^T(\mathbf{z} - \mathbf{z}_0), \quad \forall \mathbf{z} \in \mathcal{R}. \tag{3.15}$$

The set of all subgradients at $\mathbf{z}_0$ is called the subdifferential at $\mathbf{z}_0$, denoted by $\partial g(\mathbf{z}_0)$, and is always a non-empty convex compact set. A basic property associated with the subdifferential is that $\mathbf{z}_0$ is a local minimum of g if and only if $0 \in \partial g(\mathbf{z}_0)$.

Theorem 3.1. *Consider a continuous function $f(\mathbf{z})$ bounded below, such that*:

(D0) *The level set $\mathcal{R} = \{\mathbf{z}\colon\ f(\mathbf{z}) \leqslant f(\mathbf{z}_0)\}$ is compact.*

Let the proximity function $\phi(\mathbf{u}, \mathbf{v})$ satisfy, for every given $\mathbf{v}$.

(D1) $\phi(\mathbf{v}, \mathbf{v}) = 0$, $\phi(\mathbf{u}, \mathbf{v}) \geqslant 0$.
(D2) *The Hessian with respect to the first argument $\mathbf{H}_1(\mathbf{u}, \mathbf{v}) = [\partial^2\phi/\partial u_i \partial u_j]_{i,j}$ is bounded, i.e., $\|\mathbf{H}(\mathbf{u}, \mathbf{v})\| \leqslant M, \forall \mathbf{u}$.*
(D3) *ϕ is convex.*

Define the kth surrogate function f_k as

$$f_k(\mathbf{z}) = f(\mathbf{z}) + \phi(\mathbf{z}, \mathbf{z}_k), \tag{3.16}$$

and the prox-algorithm as

$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} f_k(\mathbf{z}). \tag{3.17}$$

Then any limit point $\bar{\mathbf{z}}$ of the sequence $\{\mathbf{z}_k\}$ is a stationary point of f, i.e., $0 \in \partial f(\bar{\mathbf{z}})$, and $f(\mathbf{z}_k) \to f(\bar{\mathbf{z}})$.

Proof. The proof is organized as follows: First, we show that the sequence $\{f(\mathbf{z}_k)\}$ is monotonically decreasing, hence a limit point exists. Next, we relate the difference between two consecutive iterations to the gradient's norm. Then, by assuming that the limit point is not stationary, we reach a contradiction with f being bounded below. We consequently conclude that $f(\mathbf{z}_k)$ converges to $f(\bar{\mathbf{z}})$. A full proof is presented in Appendix B. □

The functionals of (3.9) satisfy $\mathbf{H}(\mathbf{u}, \mathbf{v}) = 2(\mathbf{\Lambda} - \mathbf{A}^T\mathbf{A})$, consequently $0 < \mathbf{H}(\mathbf{u}, \mathbf{v}) \leqslant 2\max_j\{\mathbf{\Lambda}_{jj}\}$. As a result, assumptions (D0)–(D2) are trivially met. As for f from (1.1), if $\{\rho_j\}$ are continuous and strictly unimodal, then all of the conditions are fulfilled. As an example, for $\rho_j(z) = \mathbf{w}(j)|z|^p$ with $p > 0$ the theorem is valid, even for $0 < p \leqslant 1$, in which case f is not differentiable. The theorem guarantees convergence to a local minimizer, if the initial point lies inside the minimizer's neighborhood. Global convergence takes place whenever f is uniquely minimized, i.e. for $p > 1$. Also, global convergence occurs for the smoothed objective functions from (2.8).

3.3. Asymptotic convergence rate

As in [23], only the approximate convergence rate will be analyzed here. However, we will also provide specific formulae for the RLS problem (1.1). For this analysis to hold, we have to presuppose that $f \in \mathcal{C}^2$. Therefore, whenever f is not differentiable, a smoothed version, such as the one from (2.8), must be used. Now, the SSF algorithm defines a mapping $\mathbf{z}_{k+1} = M(\mathbf{z}_k)$, so we can write the first-order Taylor expansion near the optimum $\mathbf{z}^*$ as

$$\mathbf{z}_{k+1} \approx \mathbf{z}^* + \nabla M(\mathbf{z}^*)(\mathbf{z}_k - \mathbf{z}^*). \tag{3.18}$$

Assuming the approximation is exact, and by definition of the matrix norm, the asymptotic behavior conforms to

$$\|\mathbf{z}_{k+1} - \mathbf{z}^*\| \leqslant \left\|\nabla M(\mathbf{z}^*)\right\| \|\mathbf{z}_k - \mathbf{z}^*\|, \tag{3.19}$$

i.e., linear convergence rate.

In our case, the mapping $M(\cdot)$ is implicitly defined by the minimization of (3.8), which implies

$$0 = h(\mathbf{z}_{k+1}, \mathbf{z}_k) = \nabla f(\mathbf{z}_{k+1}) + \nabla_1 \phi(\mathbf{z}_{k+1}, \mathbf{z}_k). \tag{3.20}$$

Employing again the first-order Taylor expansion for h results in

$$dh = \nabla_1 h d\mathbf{z}_{k+1} + \nabla_2 h d\mathbf{z}_k = \left(\nabla^2 f + \nabla_1^2 \phi\right) d\mathbf{z}_{k+1} + \nabla_{12}^2 \phi d\mathbf{z}_k = 0. \tag{3.21}$$

This gives

$$d\mathbf{z}_{k+1} = -\left(\nabla^2 f + \nabla_1^2 \phi\right)^{-1} \nabla_{12}^2 \phi d\mathbf{z}_k, \tag{3.22}$$

i.e.,

$$\nabla M(\mathbf{z}^*) = -\left(\nabla^2 f + \nabla_1^2 \phi\right)^{-1} \nabla_{12}^2 \phi, \tag{3.23}$$

all calculated at $\mathbf{z}^*$.

Assuming the symmetry $\phi(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{v}, \mathbf{u})$ holds, then

$$\nabla M(\mathbf{z}^*) = -\left(\mathbf{I} + \left(\nabla_1^2 \phi\right)^{-1} \nabla^2 f\right)^{-1}. \tag{3.24}$$

Therefore, the asymptotic rate from (3.19) amounts to

$$\|\mathbf{z}_{k+1} - \mathbf{z}^*\| \leqslant \left(1 + \frac{\lambda_{\min}(\nabla^2 f)}{\lambda_{\max}(\nabla_1^2 \phi)}\right)^{-1} \|\mathbf{z}_k - \mathbf{z}^*\|. \tag{3.25}$$

In the case of a tight frame, i.e. $\alpha \mathbf{A}^T \mathbf{A} = \mathbf{I}$, then $\lambda_{\max}(\nabla_1^2 \phi) = 1/\alpha$.

3.4. Relation between PCD and SSF

As we have seen, although the reasoning behind PCD and SSF seems totally different, their corresponding updating rules in (2.4) and (3.10) are extremely alike. One noticeable disparity is the line-search applied by the PCD algorithm, not needed by SSF. It is clear that utilizing a line-search within SSF as well, will not damage, if not accelerate, the convergence rate. Hence, we introduce here a new algorithm comprised by (3.10) and a line-search along $\mathbf{d}_k = \mathbf{z}_{k+1} - \mathbf{z}_k$. This algorithm will be denoted hereafter by SSF-LS (stands for SSF-line-search).

Other than this, the switch $\mathbf{\Lambda} \leftrightarrow \operatorname{diag}\{\mathbf{A}^T \mathbf{A}\}$ equalizes the two formulae. Still, we cannot place $\operatorname{diag}\{\mathbf{A}^T \mathbf{A}\}$ instead of $\mathbf{\Lambda}$ in (3.10) and expect a successful algorithm. This is because the condition $\mathbf{\Lambda} \succ \mathbf{A}^T \mathbf{A}$ is not necessarily met anymore. Actually, these two matrices are not ordered in general, and thus the obtained algorithm can diverge, as occurred in our simulations. To nevertheless get an insight into how the two algorithms relate to each other, let us look at the special case where $\mathbf{A}$ is a union of L orthonormal dictionaries, formally

$$\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_L], \qquad \mathbf{A}_l^T \mathbf{A}_l = \mathbf{A}_l \mathbf{A}_l^T = \mathbf{I}, \quad \forall l. \tag{3.26}$$

In that case,

$$\left\|\mathbf{A}^T \mathbf{A}\right\| = \left\|\mathbf{A}\mathbf{A}^T\right\| = \|L\mathbf{I}\| = L. \tag{3.27}$$

Consequently, a proper choice would be $\mathbf{\Lambda} = (L + \varepsilon)\mathbf{I}$, with ε being a small positive constant. In contrast, since the columns of $\mathbf{A}$ are normalized, we have

$$\operatorname{diag}\{\mathbf{A}^T\mathbf{A}\} = \mathbf{I}. \tag{3.28}$$

This means that essentially, momentarily leaving aside the shrinkage and the line-search, a step of SSF is L-times more regularized than a step of PCD. This observation fits nicely with the fact that $\mathbf{\Lambda} = \operatorname{diag}\{\mathbf{A}^T\mathbf{A}\}$ is not large enough to ensure convergence of SSF. Regardless, experiments will determine whether this extra regularization may or may not play into the hands of SSF.

When $\mathbf{A}$ is not a union of orthonormal bases, one needs to find an appropriate diagonal matrix $\mathbf{\Lambda}$, such that $\mathbf{\Lambda} \succ \mathbf{A}^T\mathbf{A}$. Yet, in many real scenarios, little or no information is known a priori about the eigenvalues of $\mathbf{A}^T\mathbf{A}$. Calculating them may prove tedious and even impractical for long signals and redundant dictionaries. Moreover, the matrix created by placing the eigenvalues in the diagonal (i.e., diagonalization) is not assured of being as large as $\mathbf{A}^T\mathbf{A}$. Therefore, we turn to the simplest way of setting $\mathbf{\Lambda}$, which is of course $\mathbf{\Lambda} = (\lambda_{\max}\{\mathbf{A}^T\mathbf{A}\} + \varepsilon)\mathbf{I}$, with a small $\varepsilon > 0$.

To find $\lambda_{\max}$, the so-called power-method [16] may be deployed. In its most basic setting, obtaining the maximal eigenvalue of some square and positive semidefinite matrix $\mathbf{B}$ is attained by the following algorithm:

$$\mathbf{u}_k = \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}, \qquad \mathbf{v}_{k+1} = \mathbf{B}\mathbf{u}_k \quad (k = 1, 2, \ldots), \tag{3.29}$$

which yields $\lambda_{\max} = \lim_{k\to\infty} \mathbf{u}_k^T\mathbf{v}_{k+1}$. In practice, one may terminate the algorithm prematurely and multiply the obtained $\lambda_{\max}$ by a small constant, based on interpolation. This method was used throughout our experiments, whenever non-tight dictionaries were employed.

4. Acceleration using subspace optimization (SESOP)

In the previous sections we have described two recently devised element-wise optimization algorithms for solving the RLS problem, i.e., parallel coordinate descent (PCD) and separable surrogate functionals (SSF). For the latter, we suggested accelerating it by applying line-search along its descent direction, a method we call SSF-LS. This raises the question whether both methods can be further sped up, without changing their underlying structure. Apparently, the recently developed sequential subspace optimization (SESOP) method [28], holds the key for this goal. We shall briefly describe it (with slightly different notations than in [28]), focusing on its application for our objective function.

Assume that $f(\mathbf{z})$ is to be minimized via an iterative algorithm. Instead of searching along a single direction $\mathbf{r}_k$ in each iteration, a set of directions, $\{\mathbf{r}_k^i\}_{i=1}^{M+1}$, are specified. The next solution is thus obtained by minimizing f within the subspace spanned by these directions, i.e.,

$$\alpha_k = \arg\min_{\alpha} f(\mathbf{z}_k + \mathbf{R}_k\alpha), \tag{4.1}$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mathbf{R}_k\alpha_k, \tag{4.2}$$

where the columns of $\mathbf{R}_k$ are $\{\mathbf{r}_k^i\}_{i=1}^{M+1}$.

In [28] it is proved that as long as $\mathbf{R}_k$ includes the current gradient and the two so-called Nemirovski directions [29], the worst-case convergence rate of $\|f(\mathbf{z}_k) - f(\mathbf{z}^*)\|$ is optimal—$\mathcal{O}(k^{-2})$, holding true for all k. However, we are rather more interested at the frequent scenario, in which the convergence is much faster. Thus, as in [28], we discard Nemirovski-directions, leaving only the current gradient $\mathbf{r}_k^1 = \mathbf{g}_k$. Regardless, the subspace may be enriched with M previous propagation directions, specifically

$$\mathbf{r}_k^i = \mathbf{z}_{k+2-i} - \mathbf{z}_{k+1-i}, \quad i = 2, \ldots, M+1. \tag{4.3}$$

We hereby define the SESOP-M algorithm using Eqs. (4.1) and (4.2), where

$$\mathbf{R}_k = [\mathbf{g}_k, \mathbf{z}_k - \mathbf{z}_{k-1}, \ldots, \mathbf{z}_{k-M+1} - \mathbf{z}_{k-M}]. \tag{4.4}$$

As [28] shows, when f is quadratic, SESOP-1 is equivalent to the conjugate-gradients (CG) algorithm [15]. Adding more previous directions would not help, because of the expanding-manifold property of the CG, meaning that it minimizes f over the subspace spanned by the current gradient and all of the previous gradients and directions. In general, of course, f is only approximately quadratic near its minimum, thus SESOP-M with $M > 1$ may nevertheless speed up the algorithm.

The reason for expanding the search domain lies in the way f is constructed. Consider a function which can be written as $f(\mathbf{z}) = \varphi(\mathbf{Az}) + \psi(\mathbf{z})$, as our objective function from (1.1) can. Even if the product $\mathbf{Az}$ is efficiently implemented, typically its computational cost far surpasses that of $\psi(\mathbf{z})$. Therefore, matrix–vector multiplications involving $\mathbf{A}$ or $\mathbf{A}^T$ should be avoided as much as possible. This desired property is one of the main features of SESOP, as will be explained hereby. To simplify the analysis, we will refer to calculations stemming from $\varphi(\mathbf{Az})$ alone, as was justified above.

First, we note that a vector within the search-subspace is given by

$$\mathbf{z} = \mathbf{z}_k + \mathbf{R}_k\alpha = \mathbf{z}_k + \sum_{i=1}^{M+1} \alpha_i \mathbf{r}_k^i, \tag{4.5}$$

which yields

$$\mathbf{Az} = \mathbf{Az}_k + \mathbf{AR}_k\alpha = \mathbf{Az}_k + \sum_{i=1}^{M+1} \alpha_i \mathbf{Ar}_k^i. \tag{4.6}$$

Next, we can write

$$\nabla_\alpha \varphi(\mathbf{Az}) = (\mathbf{AR}_k)^T \nabla\varphi(\mathbf{Az}). \tag{4.7}$$

It means that if we calculate $\mathbf{AR}_k$ and $\mathbf{Az}_k$ at the beginning of the subspace-optimization, no further calculations (effectively) are needed during this optimization. This is true as long as $M \ll K$ (as often is the case), since $\mathbf{R}_k$ has K rows and $M+1$ columns.

Now, $\mathbf{Az}_k$ is obtained automatically from the previous subspace-optimization. In addition, in our case, M columns of $\mathbf{R}_k$ are propagation directions, for which no calculations are required, since

$$\mathbf{Ar}_k^i = \mathbf{Az}_k - \mathbf{Az}_{k-1}. \tag{4.8}$$

There are consequently only two matrix–vector multiplications per iteration, one in

$$\mathbf{r}_k^1 = \mathbf{g}_k = \mathbf{A}^T \varphi(\mathbf{Az}_k), \tag{4.9}$$

and the second is $\mathbf{Ar}_k^1$. It is the same complexity as of any gradient-descent method, yet with potentially much faster convergence. Of course, as M grows, so does the computational burden of the subspace optimization. This burden introduces a trade-off, which should be taken into consideration when utilizing this method.

Here is where we involve the element-wise optimization algorithms: Instead of choosing $\mathbf{r}_k^1 = \mathbf{g}_k$, we alternatively set $\mathbf{r}_k^1$ as the descent direction of either PCD or SSF-LS. These new algorithms will be subsequently referred to as PCD-SESOP-M and SSF-SESOP-M, respectively, with M indicating the number of previous propagation directions. These algorithm are obviously guaranteed to converge, since they do so already without the extra directions, as Sections 2.3 and 3.2 showed. The computational load remains the same as in SESOP-M, since $\mathbf{r}_k^1$ still involves only one matrix–vector multiplications, as formulae (2.4) and (3.10) show.

We are also able to provide here an asymptotic behavior analysis of PCD-SESOP-M for the quadratic case. We know from Section 2.4 that the PCD direction equals the minus of the gradient, preconditioned by $\mathbf{\Lambda} = \operatorname{diag}^{-1}\{\mathbf{H}\}$. Moreover, we mentioned earlier in the section that SESOP-M is equivalent to CG, when applied to quadratic functions. As a result, PCD-SESOP-M is actually the CG algorithm, preconditioned by $\mathbf{\Lambda}$, denoted hereafter PRE-CG. Its convergence rate thus follows [30]

$$f(\mathbf{z}_{k+1}) - f(\mathbf{z}^*) \leqslant \left(\frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}}\right)^2 \big(f(\mathbf{z}_k) - f(\mathbf{z}^*)\big), \tag{4.10}$$

m and M being the smallest and largest eigen-values of $\mathbf{\Lambda}^{-1/2}\mathbf{H}\mathbf{\Lambda}^{-1/2}$, respectively. This rate is typically much better than that of PCD (2.19).

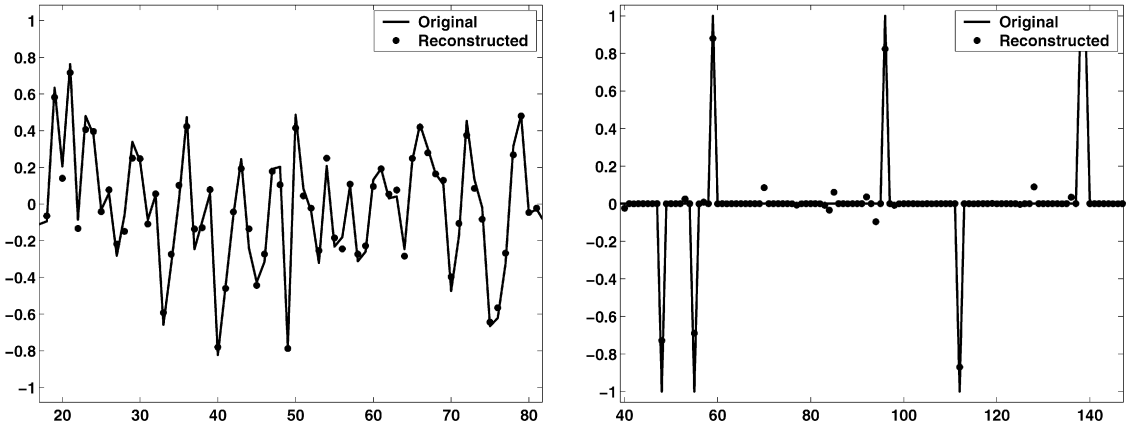

Fig. 3. Original vs reconstructed signal. Left: Time-domain. Right: Transform-domain.

5. Experimental results

5.1. Synthetic data

We have conducted denoising simulations using the BPDN formulation (1.2). The following set of experiments is performed on synthetically built 1D sparse signals of length N. The dictionary $\mathbf{A}$ is comprised of L random orthonormal bases, meaning that $K = LN$ is the length of the coefficients vector $\mathbf{z}$. We denote $\mathbf{z}_0$ as the original coefficients vector, which is built by randomly choosing a few non-zero elements (either 1 or -1) with probability γ. The clean signal $\mathbf{x}$ is then given by $\mathbf{x} = \mathbf{A}\mathbf{z}_0$, while its noisy version is $\mathbf{b} = \mathbf{x} + \mathbf{n}$, where $\mathbf{n}$ is a white-Gaussian pseudo-random noise with variance σ_n^2. Figure 3 shows part of $\mathbf{x}$ (left) and of $\mathbf{z}_0$ (right), where $N = 128$, $L = 4$ and $\gamma = 0.03$.

Since the signals in our experiments are parsimonious, we use the BPDN problem with the most sparsity-inducing prior still maintaining convexity, i.e., with $\rho(z) = \mathbf{w}(j)|z|$. To ease the optimization and to allow asymptotic behavior analysis, we shall employ the smoothed version (2.8) instead, with $s = 1/5000$. Since we have no a priori information about the non-zero elements locations, we choose constant weights $\mathbf{w}(j) = w$, with w manually-set to give the best performance (SNR-wise). Figure 3 depicts the reconstructed coefficients vector $\mathbf{z}^*$ (right) and its corresponding time-domain signal $\hat{\mathbf{x}} = \mathbf{A}\mathbf{z}^*$ (left), where $\sigma_n = 0.1$. Unless stated otherwise, all of the above parameters were used throughout our presented simulations. We have tried various settings, all giving similar results to those shown hereafter.

Before we continue, let us define a few more algorithms used in our simulations. PRE-SESOP-M stands for the SESOP-M algorithm, where the gradient direction is preconditioned by $\mathbf{\Lambda} = \operatorname{diag}^{-1}\{\mathbf{H}\}$. Another algorithm is the truncated-Newton (TN) [30], in which each iteration contains several CG steps. The effective complexity of all of the experimented algorithms is two matrix–vector multiplications per iteration, except that of TN, which is several times larger. For adequate comparison, we have measured iterations in two matrix–vector multiplications units.

Figure 4 (left) shows the function error in logarithmic scale, for CG, TN, SESOP-0,1,8, and PRE-CG. First, it is readily apparent that TN and SESOP-0 (which is actually the steepest-descent method) are much slower than the other methods, and thus will not be examined further. Secondly, the convergence rates of CG, SESOP-1 and SESOP-8 are approximately the same asymptotically, as was predicted in Section 4. The gap in favor of SESOP-8 opens up during the initial phase of optimization, where the objective function is far from being quadratic. Finally, PRE-CG was put into this figure to show its clear superiority to the not-preconditioned methods, and to therefore eliminate any reference to the previous methods in the following examples.

Figure 4 (right) depicts a comparison between PRE-CG, PRE-SESOP-1 and PRE-SESOP-8. As can be seen, the asymptotic convergence rates of these algorithms are approximately equivalent. This originates from the asymptotic equality between SESOP-M (for $M \geqslant 1$) and CG for quadratic functions (see Section 4). Any minor differences are the consequence of diversions of the objective function (2.8) from the perfect quadratical nature near the minimum. As in the previous comparison, the existing gaps open up during the first number of iterations. This demonstrates the advantage in expanding the subspace optimization within each iteration, when the function is not quadratic.

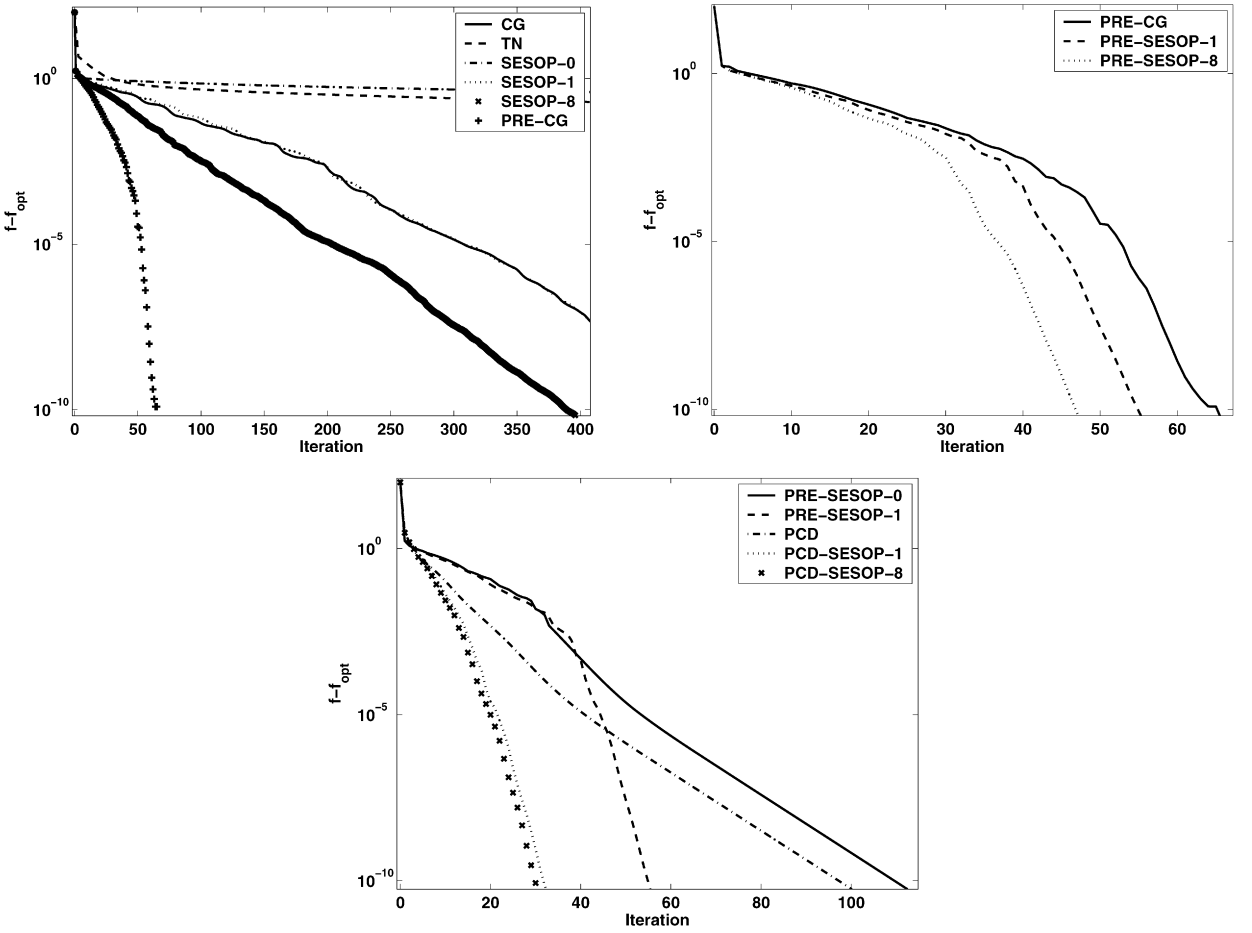

Fig. 4. Evolution of the function error in logarithmic scale, for various algorithms.

In Fig. 4 (bottom) we compare PRE-SESOP-M ($M = 0, 1$), PCD (which is the same as PCD-SESOP-0) and PCD-SESOP-M ($M = 1, 8$). To begin with, asymptotically PCD and PRE-SESOP-0 are equal, as was predicted in Section 3.3, since PRE-SESOP-0 is actually the preconditioned gradient-descent method. In addition, PRE-SESOP-M and PCD-SESOP-M (both with $M \geqslant 1$) are also asymptotically equivalent, which is expected following Section 4. However, by introducing the PCD-direction into the original PRE-SESOP-M method (for $M \geqslant 0$), we were able to greatly accelerate the optimization during the initial phase, as the figure clearly demonstrates. We have also included in this comparison the PCD-SESOP-8 method, which performs marginally better than PCD-SESOP-1, yet naturally demands more calculations per iteration. This fact, verified in many other simulations, enables us to discard PCD-SESOP-M (with $M > 1$) from future experiments.

We shall make now a numerical comparison between the theoretical convergence rate and the experimental one. The predicted lower-bound rate of PCD (also shared by PRE-SESOP-0), given by (2.19), is 0.087. It was calculated using the Hessian $\mathbf{H}$ at the minimum point $\mathbf{z}^*$. This compares with the experimental value of 0.088, calculated by measuring the asymptotic slope of the PCD graph, averaged over many noise realizations. The predicted lower-bound of PCD-SESOP-1 (also shared by PRE-SESOP-1 and PRE-CG), given by (3.25), is 0.40, compared with the experimental value of 0.46. Hence, both formulae (2.19) and (3.25) are validated in our experiments, showing that the objective function is approximately quadratic near its minimum.

The following set of simulations, summarized in Fig. 5, include PCD, PCD-SESOP-1, SSF, SSF-LS and SSF-SESOP-1. As was concluded in Section 3.4, when $\mathbf{A}$ is a union of L orthonormal bases, the SSF-direction is in some sense L-times more regularized than the PCD-direction. Hence, it is worthwhile exploring these algorithms' performance as a function of L ($L = 2, 4, 8, 16$). First, as is readily evident, SSF is asymptotically much worse than

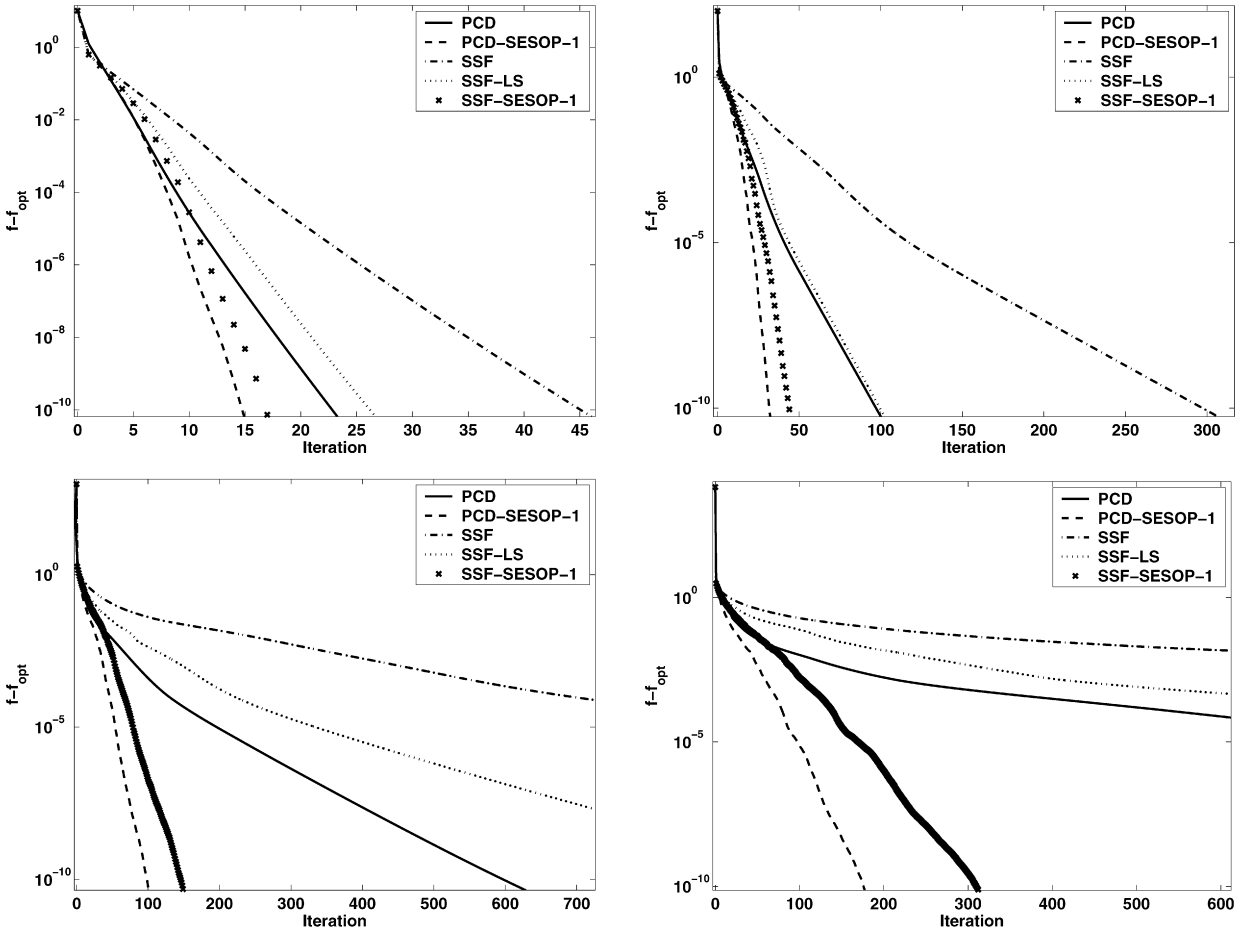

Fig. 5. Evolution of the function error in logarithmic scale for a union of L orthonormal bases. From left to right and top to bottom: $L = 2, 4, 8, 16$.

the other algorithms, specifically SSF-LS and SSF-SESOP-1. It means that our proposed techniques for accelerating SSF, i.e., line-search and expanding the subspace dimension, are indeed helpful. Now, the figure shows that PCD performs comparably or slightly better than SSF-LS asymptotically, with or without an additional search direction. All in all, the aforementioned theoretical difference between both algorithms does not seem to dramatically differentiate their asymptotic behaviors, at this particular setting. As L grows all of the algorithms are becoming slower, yet their inter-relations remain the same.

We conclude this subsection by examining the SNR improvement (or ISNR) using the original setting (i.e., $L = 4$), only that $N = 512$ rather than $N = 128$. In Fig. 6 (left), the ISNR progress of PRE-CG, PRE-SESOP-0,1,8 and PCD is displayed. The PRE-SESOP-0 method is extremely slow in achieving the ultimate ISNR value, despite being asymptotically equivalent to the PCD algorithm. Moreover, PRE-CG and PRE-SESOP-M (with $M \geqslant 1$) are also much slower than PCD, although their asymptotic rate easily surpasses that of PCD, as was shown in Fig. 4 (bottom). Incrementing M even more does not significantly improve the PRE-SESOP-M performance, still being much inferior compared to PCD. It should be noted that the first algorithms tested in this section, i.e., CG, TN and SESOP-M, are way below par and are therefore not presented.

Figure 6 (right) displays the performance of the element-wise optimization algorithms, PCD and SSF, and their improved versions, PCD-SESOP-1, SSF-LS and SSF-SESOP-1. Once again, just as demonstrated in Fig. 5, introducing line search and subspace optimization to SSF is helpful. The PCD method also profits from the extra search direction. By comparing PCD with SSF-LS and PCD-SESOP-1 with SSF-SESOP-1, i.e., equal subspace dimension, we observe the superiority of the PCD-direction. In fact, PCD-SESOP-1 exhibits both the fastest SNR increment and

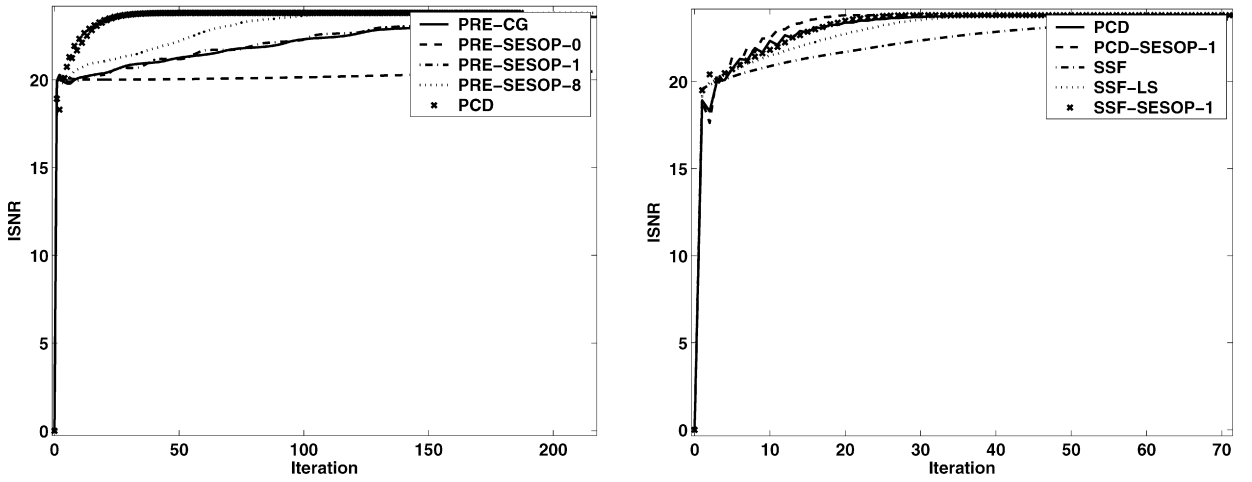

Fig. 6. The ISNR progress for various algorithms.

the highest convergence rate (see Fig. 5), and is thus the best method overall. This conclusion will be verified in the image-denoising experiments to follow.

5.2. Image denoising

The RLS problem can be used to remove noise from images as well. We shall utilize the contourlet transform (CT) [9], which is one of several transforms developed in recent years, aimed at improving the representation sparsity of images over the wavelet transform (WT) [26]. The main feature of these novel transforms (e.g., curvelets [35], bandeletes [31] and steerable wavelets [32]) is their potential to efficiently handle 2D singularities, i.e., edges, unlike wavelets which deal successfully only with point singularities. A newer version of the CT, giving better performance, was recently developed in [24], and was thus employed throughout our simulations (where $\mathbf{A}$ indicates the inverse contourlet transform). This transform is not necessarily tight (it depends on the specific filters used), and is up to $7/3$ redundant.

Experiments made on natural images show that the contourlet coefficients at different scales and directions have different average variance. Hence, the variance σ_j^2 of each coefficient should depend on the scale, the direction, and perhaps the spatial position as well. This observation justifies the existence of a weights vector $\mathbf{w}$ in the RLS formulation, rather than just a scalar. As is common in literature, the coefficients are modeled using a Laplacian distribution, $p(\mathbf{z}(j)) \propto \exp(-\frac{\sqrt{2}}{\sigma_j}|\mathbf{z}(j)|)$. This model approximates the real distribution, which is more heavy-tailed. The MAP estimator is thus given by

$$f(\mathbf{z}) = \frac{1}{2\sigma_n^2}\|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2^2 + \sqrt{2}\sum_j \big|\mathbf{z}(j)\big|/\sigma_j, \tag{5.1}$$

where σ_n^2 is the noise variance at the image domain. By comparing (1.1) (with $p = 1$) and (5.1), we get $\mathbf{w}(j) = 2\sqrt{2}\sigma_n^2/\sigma_j$. As in the 1D scenario, the smoothed version of the RLS problem (2.8) will be deployed, with $s = 1/5000$.

The implementation of this algorithm requires learning of the image prior parameters directly from the given image, i.e., estimation of the variances $\{\sigma_j^2\}$. We employ here a method introduced by Chang et al. [4] for the WT, though it remains valid for any multiscale transform like the CT. Full details and explanations of this estimation process appear in [12].

Figure 7 compares the ISNR evolution of several algorithms, when deployed to denoise the *Peppers* image, corrupted by a white-Gaussian noise of $\sigma_n = 20$ (where the dynamic range is $[0, 255]$). These result typify the various settings used, including other images and noise levels. Both comparisons repeat those made in Fig. 6 for 1D signals. The left side is meant to demonstrate the extreme superiority of PCD relative to classic optimization techniques. The right side confronts the element-wise optimization algorithms and their enhanced variants, devised in this paper. In

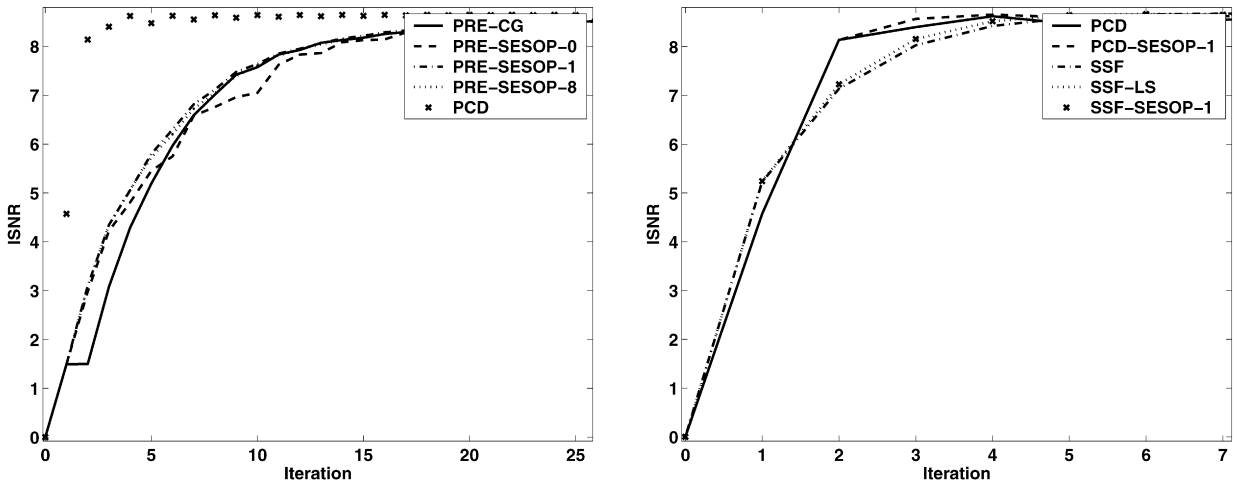

Fig. 7. Denoising of *Peppers*: ISNR progress for various algorithms.

Fig. 8. Denoising of *Peppers*. From left to right and top to bottom: Original; noisy; iteration #1; iteration #2; iteration #3; final.

essence, the results reiterate those displayed in Fig. 6, only that the superiority of the PCD family over the SSF family is augmented. Once again, the PCD-SESOP-1 proves to be the best algorithm, just as was concluded in the 1D case, although not as decisively as before.

Figure 8 shows the original *Peppers* image, its noisy version, the first three estimates using the PCD-SESOP-1 algorithm, and the final reconstruction. As seen in Fig. 7, the first two iterations just about complete the denoising process, and obtain the same improvement as PCD. Also, the second estimate is almost indistinguishable from the

Fig. 9. Denoising of *Barbara*. From left to right and top to bottom: Original; noisy; iteration #1; iteration #2; iteration #3; final.

final reconstructed image, demonstrating the practical nature of this algorithm. The denoising progress of the PCD-SESOP-1 algorithm for *Barbara* is depicted in Fig. 9, showing again the fast convergence. These results demonstrate that excellent denoising can be obtained with as little as two successive shrinkage operations. It should be emphasized however that the contourlet dictionary is not well suited for textured images like *Barbara*, and a different dictionary (or a mixture of two) may further enhance the results.

6. Conclusion

Minimizing the regularized least squares (RLS) objective function poses an important optimization task that serves various inverse problem applications. Our work explored effective ways to minimize this function using a recently introduced family of element-wise optimization algorithms. In this work we presented these algorithms, showed their near-equivalence and their inter-relations, proved their convergence, and analyzed their rate of convergence. We have also introduced a family of regularization functions, which are close to the l_p norm, and permit analytic solution for an element-wise optimization. Moreover, we have proposed a way of speeding-up these algorithms, by fusing them into the sequential-subspace-optimization (SESOP) method [28]. All of the discussed methods, as well as other classic optimization techniques, were thoroughly tested and compared in various scenarios for the denoising application. These experiments have shown the strength of the new element-wise optimization algorithms compared to well-established approaches. The PCD-SESOP-1 method, built as a merge of PCD and SESOP, proved to be the most successful one, both in minimizing the objective function and in reconstructing the desired signal. Nevertheless, all of the discussed algorithms should be tested on other inverse problems as well, in order to verify our results.

Appendix A. Proof of Theorem 2.1

By definition, $\mathbf{d}_k$ is a non-negative linear combination of descent directions, and as such, it is also a descent direction. This, combined with assumption (D0), ensures that the set of solution points, $\{\mathbf{z}_k\}_{k=0}^{\infty}$, is bounded by the compact level set $\mathcal{R}$. In the remainder of the proof, we shall refer to vectors in $\mathcal{R}$ exclusively, without specifically mentioning it. Now, every bounded series has at least one limit point, $\bar{\mathbf{z}}$ [34], which means that there exists a sub-series $\{\mathbf{z}_{k_n}\}$ converging to $\bar{\mathbf{z}}$. For notation simplicity, we denote $\mathbf{v}_n = \nabla f(\mathbf{z}_{k_n})$, and $\bar{\mathbf{v}} = \nabla f(\bar{\mathbf{z}})$.

Let us assume that $\|\bar{\mathbf{v}}\| > 0$, eventually trying to contradict one of the assumptions. This means that at least one element of $\bar{\mathbf{v}}$ satisfies $\bar{\mathbf{v}}(j) \neq 0$. Without losing generality, we can assume $\bar{\mathbf{v}}(1) \neq 0$, and even $\bar{\mathbf{v}}(1) = \varepsilon > 0$. The rest of the proof can be easily adapted to $j \neq 1$ and to $\varepsilon < 0$. From the continuity of ∇f, guaranteed by (D1), it follows that

$$\lim_{n\to\infty} \mathbf{v}_n = \nabla f\Big(\lim_{n\to\infty} \mathbf{z}_{k_n}\Big) = \bar{\mathbf{v}}. \tag{A.1}$$

Hence, there exists n_0, such that

$$\mathbf{v}_n(1) > \bar{\mathbf{v}}(1)/2 = \varepsilon/2 > 0, \quad \forall n > n_0. \tag{A.2}$$

Condition (D1) ensures that M upper limits the absolute second derivative of f. To continue from here, we state and prove the following

Lemma 1. *Consider a one-dimensional function $g \in \mathcal{C}^2$, satisfying $|g''(z)| \leqslant M, \forall z$, and suppose $g'(x) \neq 0$, $g'(y) = 0$. Then*

$$M|y - x| \geqslant \big|g'(x)\big|. \tag{A.3}$$

Proof. Since the function $f = g'$ has a bounded first derivative (with bound M), it is Lipschitz with constant M [34], which means that for all x, y, $|f(x) - f(y)| \leqslant M|x - y|$. Replacing f by g' and given that $g'(y) = 0$, we get $M|y - x| \geqslant |g'(x)|$. □

At this stage, let us define $g(\alpha) = f(\mathbf{z}_{k_n} + \alpha \mathbf{e}_1)$. It follows from (A.2) that

$$g'(0) = \partial f/\partial z_1|_{\mathbf{z}_{k_n}} = \mathbf{v}_n(1) > \varepsilon/2 > 0, \quad \forall n > n_0. \tag{A.4}$$

In addition, from (2.15) we get

$$\mathbf{d}_{k_n}(1) = \arg\min_{\alpha} g(\alpha) = \alpha_0|_{g'(\alpha_0)=0}, \tag{A.5}$$

where one point from the minima set can be arbitrarily chosen if the above minimizer is not unique. Combining (A.4) and (A.5) into Lemma 1 results in

$$\big|\mathbf{d}_{k_n}(1)\big| \geqslant \frac{\varepsilon}{2M}, \quad \forall n > n_0. \tag{A.6}$$

Now, for every coordinate, the minimization in (2.15) is done at the opposite direction of the one-dimensional derivative. It follows immediately that $\mathbf{v}_n(j)\mathbf{d}_{k_n}(j) \leqslant 0$, $\forall j$. This finally yields

$$\mathbf{v}_n^T \mathbf{d}_{k_n} = \sum_j \mathbf{v}_n(j)\mathbf{d}_{k_n}(j) \leqslant \mathbf{v}_n(1)\mathbf{d}_{k_n}(1) < -\frac{\varepsilon}{2}\frac{\varepsilon}{2M} = -\frac{\varepsilon^2}{4M} = \varepsilon_1 < 0, \quad \forall n > n_0. \tag{A.7}$$

The first inequality follows from $\{\mathbf{v}_n(j)\mathbf{d}_{k_n}(j)\}$ all being non-positive, hence taking only the first product increases the value. The second inequality results from multiplying (A.2) with (A.6), and using the aforementioned non-positivity of their product.

Let us define the function h as $h(\mu) = f(\mathbf{z} + \mu\mathbf{d})$, for all $\mu \geqslant 0$ satisfying $\mathbf{z} + \mu\mathbf{d} \in \mathcal{R}$. Since $\{\mathbf{d}(j)\}$ are obtained via a minimization problem, $\mathbf{z}(j) + \mathbf{d}(j)$ cannot exceed the level set imposed by $f(\mathbf{z})$. Hence, the entire set of directions $\{\mathbf{d}\}$ corresponding to $\mathcal{R}$ is also bounded. In other words, there exists some $R < \infty$, such that $\|\mathbf{d}\| \leqslant R$ in $\mathcal{R}$. Using this fact and the boundedness of the Hessian, we get

$$h''(\mu) = \mathbf{d}^T \mathbf{H}(\mathbf{z} + \mu\mathbf{d})\mathbf{d} \leqslant \max_{\mathcal{R}} \big\{\|\mathbf{H}\|_2 \|\mathbf{d}\|^2\big\} \leqslant MR^2. \tag{A.8}$$

We continue by stating and proving three lemmas.

Lemma 2. *Consider the one-dimensional functions $g_1, g_2 \in \mathcal{C}^1$, satisfying $g_1(x) = g_2(x)$ and $g_1'(y) \leqslant g_2'(y)$, for all $y \geqslant x$. Then $g_1(y) \leqslant g_2(y), \forall y \geqslant x$.*

Proof. Assume by contradiction that there exist $y_0 > x$, such that $g_1(y_0) > g_2(y_0)$. Since $(g_1 - g_2) \in \mathcal{C}^1$, specifically at the interval $[x, y_0]$, it follows from the mean value theorem [34] that $\exists c \in (x, y_0)$ satisfying $(g_1'(c) - g_2'(c))(y_0 - x) = (g_1(y_0) - g_2(y_0)) - (g_1(x) - g_2(x)) = g_1(y_0) - g_2(y_0)$. Therefore,

$$g_1'(c) - g_2'(c) = \frac{g_1(y_0) - g_2(y_0)}{y_0 - x} > \frac{g_2(y_0) - g_2(y_0)}{y_0 - x} = 0, \tag{A.9}$$

or $g_1'(c) > g_2'(c)$, in contradiction with $g_1(y) \leqslant g_2(y), \forall y \geqslant x$, thus proving the lemma. □

Lemma 3. *Consider a strictly convex one-dimensional quadratic function $g(x) = a + bx + cx^2$, minimized at x_0. Then $g(0) - g(x_0) = b^2/4c$.*

Proof.

$$0 = g'(x_0) = b + 2cx_0 \quad \Rightarrow \quad x_0 = -b/2c \quad \Rightarrow \tag{A.10}$$

$$g(0) - g(x_0) = a - \big(a + b(-b/2c) + c(-b/2c)^2\big) = b^2/4c. \qquad \square \tag{A.11}$$

Lemma 4. *Consider a one-dimensional function g, such that $g''(x) \leqslant C, \forall x$, and let x^* be a local minimizer of g. Then*

$$g(x_0) - g(x^*) \geqslant \frac{g'(x_0)^2}{2C}, \quad \forall x_0. \tag{A.12}$$

Proof. First, to make notations simpler, denote $h(x) = g(x + x_0)$, which yields $h^{(n)}(0) = g^{(n)}(x_0), \forall n$, and denote the minimizer of h as $x_h^* = x^* - x_0$. We will prove the lemma only for $x_0 < x^*$, which means that $h'(0) < 0$ (it is assumed here that no local maxima of h exist between 0 and x_h^*). The other case is proven similarly. We start by defining a new convex quadratic function $\tilde{h}(x) = h(0) + h'(0)x + \frac{1}{2}Cx^2$, clearly satisfying $\tilde{h}''(x) = C \geqslant h''(x), \forall x$. Since $\tilde{h}'(0) = h'(0)$, deploying Lemma 2 with $g_1 = h'$ and $g_2 = \tilde{h}'$ yields $h' \leqslant \tilde{h}'$. The same lemma, this time deployed with $g_1 = h$ and $g_2 = \tilde{h}$, serves us to show that $h \leqslant \tilde{h}$. Moreover, since $h'' \leqslant C$, then from Lemma 1, the minimizer $x^* - x_0$ of h satisfies $x_h^* \geqslant -h'(0)/C$, taking into account that $h'(0) < 0$. We notice that the minimizer of $\tilde{h}$ is $\tilde{x}_h^* = -h'(0)/C$, and thus h still declines at $\tilde{x}_h^*$. This, together with $h \leqslant \tilde{h}$, assures that $h(x_h^*) \leqslant \tilde{h}(\tilde{x}_h^*)$. Finally, it follows from Lemma 3 that

$$h(0) - h(x_h^*) \geqslant \tilde{h}(0) - \tilde{h}(\tilde{x}_h^*) = \frac{h'(0)^2}{4\frac{1}{2}C} = \frac{h'(0)^2}{2C}, \tag{A.13}$$

or equivalently,

$$g(x_0) - g(x^*) \geqslant \frac{g'(x_0)^2}{2C}. \qquad \square \tag{A.14}$$

We now redefine h as $h(\mu) = f(\mathbf{z}_{k_n} + \mu \mathbf{d}_{k_n})$, and notice that $\mathbf{z}_{k_n+1} = \mathbf{z}_{k_n} + (\arg\min_\mu h(\mu))\mathbf{d}_{k_n}$. Employing Lemma 4, with $g = h, x_0 = 0, x^* = \arg\min_\mu h(\mu)$ and $C = MR^2$ (taken from (A.8)), gives

$$f(\mathbf{z}_{k_n}) - f(\mathbf{z}_{k_n+1}) = h(0) - h(\mu_0) \geqslant \frac{h'(0)^2}{2MR^2}. \tag{A.15}$$

Furthermore, we know from (A.7) that $\mathbf{v}_n^T \mathbf{d}_{k_n} < \varepsilon_1 < 0, \forall n > n_0$. This, combined with the fact that $h'(0) = \mathbf{v}_n^T \mathbf{d}_{k_n}$, develops (A.15) into

$$f(\mathbf{z}_{k_n}) - f(\mathbf{z}_{k_n+1}) \geqslant \frac{h'(0)^2}{2MR^2} > \frac{\varepsilon_1^2}{2MR^2}, \quad \forall n > n_0. \tag{A.16}$$

Since the right side of the above is a positive constant, and since there are infinitely many $\mathbf{z}_{k_n}$'s fulfilling this relation, it follows that $f(\mathbf{z}_{k_n}) \to -\infty$. This is a clear contradiction with f being lower-bounded in $\mathcal{R}$, ensured by (D0). Therefore, the limit point $\bar{\mathbf{z}}$ is a stationary point.

It is just a short route to conclude that $f(\mathbf{z}_k) \to f(\bar{\mathbf{z}})$. Employing the continuity of f, $\mathbf{z}_{k_n} \to \bar{\mathbf{z}}$ yields $f(\mathbf{z}_{k_n}) \to f(\bar{\mathbf{z}})$. Since $\{f(\mathbf{z}_k)\}$ decreases and lower-bounded, it converges. Its limit equals that of any of its sub-series, specifically $\{f(\mathbf{z}_{k_n})\}$, and thus $f(\mathbf{z}_k) \to f(\bar{\mathbf{z}})$. □

Appendix B. Proof of Theorem 3.1

Using (D0) and (D1), it is clear that $f_k(\mathbf{z})$ has a minimizer for every k. If it is not unique, we arbitrarily choose one minimizer, making the series $\{\mathbf{z}_k\}$ defined by (3.17) well defined. By (3.16), (3.17) and (D1),

$$f(\mathbf{z}_{k+1}) + \phi(\mathbf{z}_{k+1}, \mathbf{z}_k) \leqslant f(\mathbf{z}_k), \tag{B.1}$$

i.e.

$$f(\mathbf{z}_k) - f(\mathbf{z}_{k+1}) \geqslant \phi(\mathbf{z}_{k+1}, \mathbf{z}_k) \geqslant 0. \tag{B.2}$$

Now, let us define a scalar function, $g(\alpha) = \phi(\mathbf{z}_{k+1} - \alpha \mathbf{g}_{k+1}, \mathbf{z}_k)$, where $\mathbf{g}_{k+1} = \nabla_1 \phi(\mathbf{z}_{k+1}, \mathbf{z}_k)$. This function is convex due to (D3). Its first derivative at zero is

$$g'(0) = -\mathbf{g}_{k+1}^T \nabla_1 \phi(\mathbf{z}_{k+1}, \mathbf{z}_k) = -\|\mathbf{g}_{k+1}\|^2, \tag{B.3}$$

while its second derivative follows

$$g''(\alpha) = \mathbf{g}_{k+1}^T \mathbf{H}_1(\mathbf{z}_{k+1} - \alpha \mathbf{g}_{k+1}, \mathbf{z}_k)\mathbf{g}_{k+1} \leqslant M\|\mathbf{g}_{k+1}\|^2, \quad \forall \alpha, \tag{B.4}$$

where the last inequality results from (D2). At this stage we employ Lemma 4 for g, giving

$$\phi(\mathbf{z}_{k+1}, \mathbf{z}_k) = g(0) \geqslant \frac{(-\|\mathbf{g}_{k+1}\|^2)^2}{2M\|\mathbf{g}_{k+1}\|^2} = \frac{\|\mathbf{g}_{k+1}\|^2}{2M}. \tag{B.5}$$

Combining the above inequality with (B.2) yields

$$f(\mathbf{z}_k) - f(\mathbf{z}_{k+1}) \geqslant \frac{\|\mathbf{g}_{k+1}\|^2}{2M}, \tag{B.6}$$

which is the key element in our proof.

Since $\{f(\mathbf{z}_k)\}$ is monotonically-decreasing by (B.2), and because the level set $\mathcal{R} = \{\mathbf{z}\colon\ f(\mathbf{z}) \leqslant f(\mathbf{z}_0)\}$ is compact by assumption, it follows that $\{\mathbf{z}_k\}_{k=0}^{\infty}$ is bounded in $\mathcal{R}$. Therefore, there exists a sub-series $\{\mathbf{z}_{k_n}\}$ converging to a limit point $\bar{\mathbf{z}}$. Currently, we assume to the contrary that $\bar{\mathbf{z}}$ is not stationary, i.e., $0 \notin \partial f(\bar{\mathbf{z}})$. It is known that $\partial f(\mathbf{z})$ is always a non-empty convex compact set [17], which means that $\min \|\partial f(\mathbf{z})\|$ always exists. This, combined with the fact that $0 \notin \partial f(\bar{\mathbf{z}})$, ensures that $\min \|\partial f(\bar{\mathbf{z}})\| > 0$. Let us denote $\varepsilon = \min \|\partial f(\bar{\mathbf{z}})\|$. Since $\mathbf{z}_{k_n} \to \bar{\mathbf{z}}$, then there are infinitely many k_n's satisfying

$$\min \big\|\partial f(\mathbf{z}_{k_n+1})\big\| > \frac{\varepsilon}{2}. \tag{B.7}$$

By definition of the prox-algorithm,

$$0 \in \partial f(\mathbf{z}_{k_n+1}) + \nabla \phi(\mathbf{z}_{k_n+1}, \mathbf{z}_{k_n}), \tag{B.8}$$

meaning that $-\mathbf{g}_{k_n+1} \in \partial f(\mathbf{z}_{k_n+1})$, and thus according to (B.7), $\|\mathbf{g}_{k_n+1}\| > \varepsilon/2$. Plugging this relation into (B.6) gives

$$f(\mathbf{z}_k) - f(\mathbf{z}_{k+1}) > \frac{\varepsilon^2}{8M}, \tag{B.9}$$

for infinitely many k_n's, which of course negates f being bounded below. This shows that the limit point $\bar{\mathbf{z}}$ is actually stationary. The conclusion that $f(\mathbf{z}_k) \to f(\bar{\mathbf{z}})$ follows immediately, just as in the last paragraph of Theorem 2.1's proof. □

References

[1] D. Andrews, C. Mallows, Scale mixtures of normal distributions, J. Roy. Statist. Soc. 36 (1974) 99–102.
[2] A. Ben-Tal, M. Zibulevsky, Penalty/barrier multiplier methods for convex programming problems, SIAM J. Optim. 7 (2) (1997) 347–366.

[3] J.M. Bioucas-Dias, Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors, IEEE Trans. Image Process. 15 (4) (2006) 937–951.
[4] S.G. Chang, B. Yu, M. Vetterli, Spatially adaptive wavelet thresholding with context modeling for image denoising, IEEE Trans. Image Process. 9 (9) (2000) 1522–1531.
[5] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, SIAM Rev. 43 (1) (2001) 59–129.
[6] P.L. Combettes, V.R. Wajs, Signal recovery by proximal forward–backward splitting, SIAM J. Multiscale Model. Simul. 4 (4) (2005) 1168–1200.
[7] I. Daubechies, M. Defrise, C. De-Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math. LVII (2004) 1413–1457.
[8] A. Dempster, N. Laird, D. Rubin, Maximum likelihood estimation from incomplete data via the EM algorithm, J. Roy. Statist. Soc. B 39 (1977) 1–38.
[9] M.N. Do, M. Vetterli, The contourlet transform: An efficient directional multiresolution image representation, IEEE Trans. Image Process. 14 (2) (2005) 2091–2106.
[10] D.L. Donoho, De-noising by soft thresholding, IEEE Trans. Inform. Theory 41 (3) (1995) 613–627.
[11] M. Elad, Why simple shrinkage is still relevant for redundant representations? IEEE Trans. Inform. Theory 52 (12) (2006) 5559–5569.
[12] M. Elad, B. Matalon, M. Zibulevsky, Image denoising with shrinkage and redundant representations, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, New York, June 17–22, 2006, pp. 1924–1931.
[13] M.A. Figueiredo, R.D. Nowak, An EM algorithm for wavelet-based image restoration, IEEE Trans. Image Process. 12 (8) (2003) 906–916.
[14] M.A. Figueiredo, R.D. Nowak, A bound optimization approach to wavelet-based image deconvolution, in: IEEE International Conference on Image Processing, 2005.
[15] P.E. Gill, W. Murray, M.H. Wright, Practical Optimization, Academic Press, London, 1981.
[16] G.H. Golub, C.F.V. Loan, Matrix Computations, Johns Hopkins Univ. Press, Baltimore, 1996.
[17] J.P. Hiriart-Urruty, C. Lemarichal, Fundamentals of Convex Analysis, Springer, 2001.
[18] D. Hunter, K. Lange, A tutorial on MM algorithms, Amer. Statist. 58 (2004) 30–37.
[19] M. Jansen, Noise Reduction by Wavelet Thresholding, Springer, New York, 2001.
[20] A.C. Kak, M. Slaney, Principles of Computerized Tomographic Imaging, Society of Industrial and Applied Mathematics, 2001.
[21] L. Landweber, An iterative formula for Fredholm integral equations of the first kind, Amer. J. Math. 73 (1951) 615–624.
[22] M. Lang, H. Guo, J.E. Odegard, Noise reduction using undecimated discrete wavelet transform, IEEE Signal Process. Lett. 3 (1) (1996) 10–12.
[23] K. Lange, D.R. Hunter, I. Yang, Optimization transfer using surrogate objective functions (with discussion), J. Comput. Graph. Statist. 9 (1) (2000) 1–59.
[24] Y. Lu, M.N. Do, A new contourlet transform with sharp frequency localization, in: Proc. of IEEE International Conference on Image Processing, Atlanta, 2006.
[25] Z.Q. Luo, P. Tseng, On the convergence of the coordinate descent method for convex differentiable minimization, J. Optim. Theory Appl. 72 (1) (1992) 7–35.
[26] S. Mallat, A Wavelet Tour of Signal Proc, Academic Press, San Diego, CA, 1998.
[27] P. Moulin, J. Liu, Analysis of multiresolution image denoising schemes using generalized-Gaussian and complexity priors, IEEE Trans. Inform. Theory 45 (3) (1999) 909–919.
[28] G. Narkiss, M. Zibulevsky, Sequential subspace optimization method for large-scale unconstrained optimization, technical report CCIT No. 559, Technion, The Israel Institute of Technology, Haifa, 2005.
[29] A. Nemirovski, Orth-method for smooth convex optimization (in Russian), Izv. AN SSSR, Ser. Tekhnicheskaya Kibernetika 2 (1982) (the journal is translated to English as Engineering Cybernetics. Soviet J. Computer & Systems Sci.).
[30] J. Nocedal, S. Wright, Numerical Optimization, Springer, New York, 1999.
[31] E.L. Pennec, S. Mallat, Sparse geometric image representation with bandelets, IEEE Trans. Image Process. 14 (2005) 423–438.
[32] J. Portilla, V. Strela, M.J. Wainwright, E.P. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, IEEE Trans. Image Process. 12 (11) (2003) 1338–1351.
[33] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim. 14 (5) (1976) 877–898.
[34] M. Spivak, Calculus, third ed., Publish or Perish, 1994.
[35] J.L. Starck, E.J. Candes, D.L. Donoho, The curvelet transform for image denoising, IEEE Trans. Image Process. 11 (6) (2002) 670–684.
[36] O. Strand, Theory and methods related to the singular-function expansion and Landweber's iteration for integral equations of the first kind, SIAM J. Numer. Anal. 11 (1974) 798–825.