WAVELET DECOMPOSITION METHOD FOR $L_2$/TV-IMAGE DEBLURRING

M. FORNASIER\(^*\), Y. KIM\(^†\), A. LANGER\(^‡\), AND C.-B. SCHÖNLIB\(^§\)

Abstract. In this paper, we show additional properties of the limit of a sequence produced by the subspace correction algorithm proposed by Fornasier and Schönlieb [25] for $L_2$/TV-minimization problems. An important but missing property of such a limiting sequence in [25] is the convergence to a minimizer of the original minimization problem, which was obtained in [24] with an additional condition of overlapping subdomains. We can now determine when the limit is indeed a minimizer of the original problem. Inspired by the work of Vonesch and Unser [35], we adapt and specify this algorithm to the case of an orthogonal wavelet space decomposition for deblurring problems and provide an equivalence condition to the convergence of such a limiting sequence to a minimizer. We also provide a counterexample of a limiting sequence by the algorithm that doesn’t converge to a minimizer, which shows the necessity of our analysis of the minimizing algorithm.

Key words. Image deblurring, wavelet decomposition method, convex optimization, oblique-thresholding, total variation minimization, alternating minimization

AMS subject classifications. 65K10, 65M32, 49M27, 68U10, 90C25, 49J40, 42C40

1. Introduction. In image processing, one is interested in the restoration of an observed image, which is corrupted by a measurement device. Let $\Omega = [0,1]^2$ and $T : L_2(\Omega) \to L_2(\Omega)$ be a blur operator modelled as a convolution $Tu = u * \kappa$, with kernel $\kappa \in L_1(\Omega)$. Then the ideal observed noiseless image $\tilde{g}$ can be described as

$$\tilde{g} = Tu,$$

where $u \in L_2(\Omega)$ is the unknown image, which we would like to reconstruct. In general, the observed data is additionally corrupted by noise $e$, i.e.,

$$g = Tu + e. \quad (1.1)$$

We are in particular interested in the recovery of $u$ from the given noisy observed image $g$ when the operator $T$ is not invertible or ill-conditioned, and regularization techniques are required [19].

Images can be well approximated using the superposition of few wavelets [15, 30]. Hence we make the realistic assumption that $u$ can be represented by a sparse wavelet expansion, i.e., for a given wavelet basis $\{\psi_\lambda : \lambda \in \Lambda\}$ indexed by a countable set $\Lambda$ the image $u$ can be well approximated by a series expansion with few nonvanishing coefficients of the form

$$u \approx Su_\Lambda = \sum_{\lambda \in \Lambda} u_\lambda \psi_\lambda,$$

where $u_\Lambda = (u_\lambda)_{\lambda \in \Lambda} \in \ell_2(\Lambda)$ and $S : \ell_2(\Lambda) \to L_2(\Omega)$ is a bounded linear operator, called the synthesis operator. It is acknowledged that the simultaneous minimization of the least-squares discrepancy to data and of the $\ell_1$-norm of coefficients promotes sparsity [16]. Hence we consider the minimization of the functional

$$J(u_\Lambda) = \|Au_\Lambda - g\|_{L_2(\Omega)}^2 + 2\alpha \|u_\Lambda\|_{\ell_1(\Lambda)} = \|Au_\Lambda - g\|_{L_2(\Omega)}^2 + 2\alpha \sum_{\lambda \in \Lambda} |u_\lambda| \quad (1.2)$$

with respect to the vector of wavelet coefficients $u_\Lambda = (u_\lambda)_{\lambda \in \Lambda}$, where $\alpha > 0$ is a fixed regularization parameter, and $A = T \circ S : \ell_2(\Lambda) \to L_2(\Omega)$ is the composition of the synthesis map $S$ and the

*Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenbergerstrasse 69, A-4040, Linz, Austria Email: massimo.fornasier@oeaw.ac.at
†Department of Mathematics, University of California, Irvine, 410X, Rowland Hall, Irvine, CA 92697-3875, USA Email: yuno1123@math.uci.edu
‡Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenbergerstrasse 69, A-4040, Linz, Austria Email: andreas.langer@oeaw.ac.at
§Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, United Kingdom Email: C.B.Schoenlieb@damtp.cam.ac.uk
operator $T$. In order to address this minimization with respect to $u_\Lambda$, one can use, for instance, the so-called iterative soft-thresholding algorithm [16]: pick an initial $u_\Lambda^{(0)} \in \ell_2(\Lambda)$ and iterate

$$u_\Lambda^{(n+1)} = S_\alpha(u_\Lambda^{(n)} + A^*(g - A u_\Lambda^{(n)})), \quad n \geq 0,$$

(1.3)

where $S_\alpha : \ell_2(\Lambda) \rightarrow \ell_2(\Lambda)$ is defined componentwise by $S_\alpha(v) = (S_\alpha v)_\lambda \in \Lambda$, and

$$S_\alpha(v) = \begin{cases} v - \text{sign}(v)\alpha & |v| > \alpha \\ 0 & \text{otherwise} \end{cases}$$

is the so-called soft-thresholding operator. The strong convergence of the algorithm in 1.3 to minimizers of $J$ is proved in [16]. In [5] it was shown that under additional conditions on the operator $A$ or on minimizers of 1.2 the algorithm in 1.3 converges linearly, although with a rather poor rate in general, see [23] for a broader discussion. There exist several alternative approaches, that promise to solve $\ell_1$-minimization with fast convergence [17, 21, 28, 3]. One way to accelerate the speed of convergence of minimizing iterative soft-thresholding algorithms for large-scale problems was proposed in [22]. There a domain decomposition method for $\ell_1$-norm minimization was introduced and analyzed. The main idea of this algorithm is to decompose the index set $\Lambda$ into two (or more) disjoint sets $\Lambda_i$, $i = 1, 2, \ldots$, such that $\Lambda = \Lambda_1 \cup \Lambda_2$. Associated with this decomposition we define $V_i = \{u_\Lambda \in \ell_2(\Lambda) : \text{supp}(u_\Lambda) \subset \Lambda_i\}$ for $i = 1, 2$. Then we minimize J by using the following alternating algorithm: pick an initial $V_1 \oplus V_2 \ni u_\Lambda^{(0)} + u_\Lambda^{(0)} = u_\Lambda^{(0)}$, for example $u^{(0)} = 0$, and iterate

$$\begin{cases} u_{\Lambda_1}^{(n+1)} \approx \arg \min_{u_{\Lambda_1} \in V_1} J(u_{\Lambda_1}^{(n)} + u_{\Lambda_2}^{(n)}) \\ u_{\Lambda_2}^{(n+1)} \approx \arg \min_{u_{\Lambda_2} \in V_2} J(u_{\Lambda_1}^{(n)} + u_{\Lambda_2}^{(n+1)}) \\ u_{\Lambda}^{(n+1)} := u_{\Lambda_1}^{(n+1)} + u_{\Lambda_2}^{(n+1)}, \end{cases}$$

(1.4)

where $u_{\Lambda_i}$ is supported on $\Lambda_i$ only, $i = 1, 2$. We observe that the $\ell_1$-norm splits additively

$$\|u_{\Lambda_1} + u_{\Lambda_2}\|_{\ell_1(\Lambda)} = \|u_{\Lambda_1}\|_{\ell_1(\Lambda_1)} + \|u_{\Lambda_2}\|_{\ell_1(\Lambda_2)},$$

and hence the subproblems in (1.4) are of the same kind as the original problem (1.2), i.e., for example for the problem on $\Lambda_1$ we have

$$\arg \min_{u_{\Lambda_1} \in V_1} J(u_{\Lambda_1}^{(n)} + u_{\Lambda_2}^{(n)}) = \arg \min_{u_{\Lambda_1} \in V_1} \|A_{\Lambda_1} u_{\Lambda_1} - (g - A_{\Lambda_2} u_{\Lambda_2}^{(n)})\|_{L_2(\Omega)}^2 + 2\alpha \|u_{\Lambda_1}\|_{\ell_1(\Lambda_1)},$$

where $A_{\Lambda_i}$ is the restriction of the matrix $A$ to the columns indexed by $\Lambda_i$. Therefore, for solving the subminimization problems of (1.4) we can use one of the before mentioned methods, for example again the iterative thresholding algorithm:

$$u_{\Lambda_i}^{(t+1,n+1)} = S_\alpha(u_{\Lambda_i}^{(t,n+1)} + A_{\Lambda_i}^* ((g - A_{\Lambda_1} u_{\Lambda_1}^{(n)}) - A_{\Lambda_1} u_{\Lambda_i}^{(t,n+1)})), \quad i \in \{1, 2\} \setminus \{i\}.$$

(1.5)

Great advantages of this domain decomposition algorithm are that we can solve instead of one large problem several smaller problems, which might lead to an acceleration of convergence with a reduction of overall computational cost, and that it can be easily parallelized. Convergence of both the sequential and the parallel versions of this algorithm is proven in [22]. The same method was used in [35] by Vonesch and Unser with minor modifications, specifically by using Haar wavelets for deblurring (or deconvolution) problems, where cyclic updates of the different resolution levels were combined with the preconditioning effect of subband-specific parameters. The effectiveness of this method was shown by solving multidimensional image deconvolution problems, as 3D fluorescence microscopy. We give a brief and intuitive explanation of the reason why this multilevel method works so well for deblurring problems: wavelet space decompositions split the function space into orthogonal subspaces $V_i$. Note that $T$ is just a convolution operator with kernel $\kappa$ or a multiplier $\kappa$ in the Fourier domain, where the $V_i$'s represent nearly disjoint dyadic subbands, and we have
that all \( A_{\Lambda_i} \) are also nearly orthogonal, i.e., \( A_{\Lambda_i}^* A_{\Lambda_i} \approx 0 \) for \( i \neq \hat{i} \). Hence each subiteration (1.5) of the algorithm in (1.4) is (nearly) restricted to one of the \( \mathcal{V}_i \), independent of other subiterations, and converges fast as \( A_{\Lambda_i}^* A_{\Lambda_i} \) is a well-conditioned operator. This is the case whenever the Fourier transform \( \hat{\kappa} \) is, for example, a slowly decaying function on the subband associated with \( \mathcal{V}_i \), see Figure 1.1.

![Figure 1.1](image)

**Fig. 1.1.** We depict a slowly decaying envelope of the Fourier transform \( \hat{\kappa} \) of a kernel \( \kappa \). The spaces \( \mathcal{V}_1 \) and \( \mathcal{V}_2 \) are two orthogonal spaces, obtained by a wavelet decomposition and associated to nearly disjoint subbands. Restricted on the subband associated to \( \mathcal{V}_i \), the function \( \hat{\kappa} \), essentially representing the spectrum of the matrix \( A_{\Lambda_i} \), can be intuitively understood as bounded from above and below, providing the well-conditioning of the operator \( A_{\Lambda_i}^* A_{\Lambda_i} \).

To gain maximal performance of the algorithm in (1.4) we need to introduce preconditioner constants for each subiteration respectively, i.e., instead of considering \( I - A_{\Lambda_i}^* A_{\Lambda_i} \) we take iteration operators

\[
I - \frac{1}{\alpha_i} A_{\Lambda_i}^* A_{\Lambda_i},
\]

for \( \alpha_i \geq \| A_{\Lambda_i} \|^2 \).

The main goal of this paper is to transpose these observations on preconditioning effects of alternating algorithms based on wavelet decompositions to the deblurring model where the term \( \| u \|_{\ell^1(\Lambda)} \) in (1.2) is substituted by the total variation of the function \( u \). We recall that for \( u \in L^1(\Omega) \)

\[
V(u, \Omega) := \sup \left\{ \int_{\Omega} u \text{div} \varphi \, dx : \varphi \in [C_c^1(\Omega)]^2, \| \varphi \|_\infty \leq 1 \right\}
\]

is the variation of \( u \). Moreover, \( u \in BV(\Omega) \), the space of bounded variation functions [1, 20] if and only if \( V(u, \Omega) < \infty \). In this case, we denote \( |Du|(\Omega) = V(u, \Omega) \) the total variation of the finite Radon measure \( Du \), the derivative of \( u \) in the distributional sense. The space \( BV(\Omega) \) endowed with the norm \( \| u \|_{BV(\Omega)} := \| u \|_{L^1(\Omega)} + |Du|(\Omega) \) is a Banach space. The minimization of the total variation is a well-understood regularization for preserving edges of images. Rudin, Osher, and Fatemi [33] proposed the minimization of functionals with total variation constraints as a regularization technique for image denoising. From this pioneering work, total variation minimization became a standard tool in image processing, also for more sophisticated problems, such as deblurring, superresolution, inpainting etc. [2, 9, 10, 18, 34]. We also refer to [11] for an extensive introduction to the use of total variation in imaging.

Our reason for expecting that the preconditioning effects observed by Vonesch and Unser [35] for Haar wavelet-based regularization will take place also in total variation regularization of deblurring problems stems from the well-known near characterization of \( BV \) in terms of wavelets...
the BV-norm of a bivariate function $u$ is in fact nearly equivalent to the $\ell_1$-norm of its bivariate Haar wavelet coefficients $u_\Lambda$. More precisely, there exist constants $c_1, c_2 \in \mathbb{R}^+$ such that
\[ c_1 \|u_\Lambda\|_{\ell_1} + \|Du\|_1(\Omega) \leq \|u\|_{L^1(\Omega)} + |Du|_1(\Omega) \leq c_2 \|u_\Lambda\|_{\ell_1}, \]for all $u \in BV(\Omega), (1.6)$

and for all $\delta > 0$. Actually these inequalities result in embeddings of $BV$ with respect to suitable Besov spaces:
\[ B^{1,1}_1 \subset BV \subset B^{1,w}_{1,1}. \]

We refer the interested reader to [14] for more details.

Because of this observation and the above mentioned preconditioning mechanism for a deblurring operator in connection with a wavelet space decomposition, we are interested in the minimization of the functional
\[ J(u) = \|Tu - g\|_{L^2(\Omega)}^2 + 2\alpha |Du|_1(\Omega), (1.7) \]

by using a suitably adapted wavelet-based multilevel algorithm.

1.1. Our approach. Domain decomposition and subspace correction methods for functionals of the form (1.7) were already proposed in [24, 25]. There some of the authors of this paper mainly focused on the splitting of the physical domain $\Omega$ into smaller subdomains $\Omega = \bigcup_i \Omega_i$ and studied an alternating minimization algorithm on each subspace. Nevertheless, the validity of the algorithm proposed in [25] is not restricted in principle to orthogonal decompositions of the space resulting from splittings of the physical domain $\Omega$, but can also be applied to more abstract orthogonal decompositions of the function space, e.g., a wavelet space decomposition as we have it in mind here.

Let $\phi$ be a scaling function generating a multiresolution analysis $(V_i)_{i \in \mathbb{Z}}$ and $\psi$ a corresponding wavelet function. Then we obtain
\[ L^2(\Omega) = \bigcup_{i \in \mathbb{Z}} V_i = V_0 \oplus \bigoplus_{j=1}^{\infty} W_j = \bigoplus_{j \in \mathbb{Z}} W_j, \]

where $W_j$ is the wavelet space corresponding to the $j$-th level generated by the basis
\[ \{\psi_\lambda : \lambda \in \Lambda_j\}, \]

and $\Lambda_j$ denotes the set of indices for the $j$-th level, see [12, 15] for more details. Moreover, $W_j$ is the orthogonal complement of $V_j$ in $V_{j+1}$, i.e., we have
\[ V_{j+1} = V_j \oplus W_j. (1.8) \]

In particular we may decompose $L^2(\Omega)$ in the following way
\[ L^2(\Omega) = V_0 \oplus V_0^\perp = V_0 \oplus \left( \bigoplus_{j=0}^{\infty} W_j \right) \]

and denote $V_1 := V_0$ and $V_2 := V_0^\perp = \bigoplus_{j=0}^{\infty} W_j$. Associated with this wavelet decomposition into two subspaces the minimization of (1.7) can be carried out by the alternating subspace correction method proposed in [25], which reads as follows: pick an initial $V_1 \oplus V_2 \ni u_1^{(0)} + u_2^{(0)} := u^{(0)}$, for example $u^{(0)} = 0$, and iterate
\[ \begin{cases} u_1^{(n+1)} \approx \arg \min_{u_1 \in V_1} J(u_1 + u_2^{(n)}) \\ u_2^{(n+1)} \approx \arg \min_{u_2 \in V_2} J(u_1^{(n+1)} + u_2) \\ u^{(n+1)} := u_1^{(n+1)} + u_2^{(n+1)}. \end{cases} (1.9) \]
In [25] an implementation of this algorithm was suggested, which guaranteed to decrease the objective energy $J$ monotonically. However, convergence to minimizers of $J$ could be proven only under technical conditions, which are in general not fulfilled, as also illustrated by numerical examples in [25].

In this paper we show additional properties of the limit of a sequence produced by the algorithm in (1.9) and obtain an additional condition under which the obtained limit is indeed the expected minimizer. Nevertheless, this condition cannot be ensured to hold always for any operator $T$. In particular, we are able to construct a counterexample, which shows that in general we cannot expect convergence of the algorithm in (1.9) to a minimizer of $J$, even for the simplest case of the identity operator $T = I$. Despite this quite special negative result, we show in this paper that an orthogonal wavelet space decomposition for deblurring problems works in practice very efficiently, as already observed by Vonesch and Unser in their study related to $\ell_1$-regularization [35]. In particular, with the help of the newly obtained condition of convergence, we are able to show in our numerical examples that the sequence produced by this algorithm in fact numerically converges to a minimizer of $J$.

Throughout the paper we eventually work on a finite dimensional space by considering a finite regular mesh as a discretization of $\Omega$. Hence we consider instead of the continuous functional (1.7) its discrete approximation, for ease again denoted by $J$ in (3.1). Note that the discrete approximation (3.1) $\Gamma$-converges to the continuous functional (1.7) (see [4]), and has the same singular nature as the continuous problem. For simplicity we will limit ourselves to decompose our problem only into two orthogonal subspaces $V_1$ and $V_2$, which is by no means a restriction, as a generalization to a multiple decomposition is straightforward, see [25, Remark 5.3]. However, we stress also that in our numerical experiments the beneficial effect of preconditioning seems not to improve significantly by considering multiple decompositions, see Section 6.

The paper is organized as follows. The main notations used throughout the paper are given in Section 2. In Section 3 we describe the algorithm in (1.9), specified to a wavelet space decomposition. The convergence of the algorithm to a minimizer of $J$ is investigated in Section 4, where we show properties of the limit of the sequence produced by the algorithm. Additionally we construct a counterexample to show that convergence cannot be obtained in general. Section 5 contains the proof of the main results. In Section 6, we show numerical examples for total variation deblurring which illustrate our findings.

2. Notations. Since we are mainly interested in image deblurring problems, it is sufficient for us to introduce our main notations for a discretization in $[0,1]^2$ only. We assume now that $\Omega$ is a 2-dimensional mesh in $[0,1]^2$ of size $N_1 \times N_2$, where $N_1, N_2 \in \mathbb{N}$. The considered function space is $H = \mathbb{R}^{N_1 \times N_2}$, with corresponding norm

$$
\|u\|_H = \|u\|_2 = \left(\sum_{x \in \Omega} |u(x)|^2\right)^{1/2}.
$$

Then the discrete gradient $\nabla u$ is the vector of the finite differences on the mesh, given by

$$(\nabla u)(x) = ((\nabla u)^1(x), (\nabla u)^2(x))$$

where

$$(\nabla u)^1(x_{i,j}) = \begin{cases} u(x_{i+1,j}) - u(x_{i,j}) & \text{if } i < N_1 \\ 0 & \text{if } i = N_1, \end{cases}$$

and

$$(\nabla u)^2(x_{i,j}) = \begin{cases} u(x_{i,j+1}) - u(x_{i,j}) & \text{if } j < N_2 \\ 0 & \text{if } j = N_2, \end{cases}$$
for $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$. Then the discrete total variation of $u$ is defined by
\[
|\nabla u|(\Omega) := \sum_{x \in \Omega} |\nabla u(x)|.
\]
where $|g| = \sqrt{y_1^2 + y_2^2}$ for every $g = (y_1, y_2) \in \mathbb{R}^2$.

For an operator $Q$ we denote by $Q^*$ its adjoint. Further we introduce the discrete divergence $\text{div} : \mathcal{H}^2 \to \mathcal{H}$ defined, in analogy with the continuous setting, by $\text{div} = -\nabla^* (\nabla^*$ is the adjoint of the discrete gradient $\nabla$). The discrete divergence operator is explicitly given by
\[
(\text{div} p)(x_{i,j}) = \begin{cases} 
p^1(x_{i,j}) - p^1(x_{i-1,j}) & \text{if } 1 < i < N_1 \\
p^1(x_{i,j}) & \text{if } i_1 = 1 \\
-p^1(x_{i-1,j}) & \text{if } i_1 = N_1 \\
p^2(x_{i,j}) - p^2(x_{i,j-1}) & \text{if } 1 < j < N_2 \\
p^2(x_{i,j}) & \text{if } j = 1 \\
-p^2(x_{i,j-1}) & \text{if } j = N_2,
\end{cases}
\]
for every $p = (p^1, p^2) \in \mathcal{H}^2$. Further we define the closed convex set
\[
K := \{ (\text{div} p : p \in \mathcal{H}^2, |p(x)| \leq 1 \text{ for all } x \in \Omega) \},
\]
where $|p(x)| = \sqrt{(p^1(x))^2 + (p^2(x))^2}$, and denote $P_K(u) = \arg \min_{v \in K} \|u - v\|_2$ the orthogonal projection onto $K$. We will also denote by $\langle \cdot, \cdot \rangle_{\mathbb{R}^2}$ the scalar product in $\mathbb{R}^2$.

3. Description of the Algorithm.

3.1. Preconditioning. We are interested in solving by the multilevel algorithm in (1.9) the minimization of the discrete functional $\mathcal{J} : \mathcal{H} \to \mathbb{R}$ defined by
\[
\mathcal{J}(u) = \|T u - g\|_2^2 + 2\alpha|\nabla u|(\Omega),
\]
where $T : \mathcal{H} \to \mathcal{H}$ is a blur operator with kernel $\kappa$, $g \in \mathcal{H}$ is a given datum, and $\alpha > 0$ is a fixed regularization parameter. Furthermore, it is convenient that we assume $\|T\| < 1$, which is not a restriction, as a proper rescaling of the problem yields the desired setting, and does not change the minimization problem. In order to guarantee the existence of minimizers for (3.1) we assume that $\mathcal{J}$ is coercive in $\mathcal{H}$, i.e., there exists a constant $C > 0$ such that $\{ u \in \mathcal{H} : \mathcal{J}(u) \leq C \}$ is nonempty and bounded in $\mathcal{H}$. It is well known that if $1 \notin \text{ker}(T)$ then this coercivity condition is satisfied, see [34, Proposition 3.1]. In addition, if $T$ is injective, for instance, if $\kappa$ is a Gaussian or an averaging convolution kernel (see Section 6), then (3.1) has unique minimizer.

We can identify $\mathcal{H}$ with the sequences of samples $(u(x))_{x \in \Omega}$ of a function $u$ on $[0,1]^2$, and with $V_1$, the first scaling space of a multiresolution analysis, by means of the map $(u(x))_{x \in \Omega} \mapsto \sum_{\lambda \in \Lambda} u(x) \varphi_{1,\lambda}$, where $\varphi_{1,\lambda}$ is a properly dilated scaling function, and $(x_\lambda)_{\lambda \in \Lambda}$ is a suitable rearrangement of the nodes of the mesh $\Omega$. Moreover, by property (1.8), we have the orthogonal splitting $\mathcal{H} = V_1 = V_0 \oplus W_0$. Of course, we may obtain further levels of decomposition.

For simplicity we restrict ourselves to a decomposition into two subspaces $\mathcal{V}_1 := V_0$ and $\mathcal{V}_2 := W_0$ only. We define
\[
\pi_{\mathcal{V}_i} : \mathcal{H} \to \mathcal{V}_i,
\]
the orthogonal projection onto $\mathcal{V}_i$, for $i = 1, 2$. Then every $u \in \mathcal{H}$ has a unique representation $u = \pi_{\mathcal{V}_1}(u) + \pi_{\mathcal{V}_2}(u)$. In the sequel we denote $u_i = \pi_{\mathcal{V}_i}(u)$, for $i = 1, 2$. Moreover we introduce surrogate functionals on $\mathcal{V}_1 \oplus \mathcal{V}_2$ for $a \in \mathcal{V}_1$ and for $i = 1, 2$ by
\[
\mathcal{J}_i(u_1, u_2; a) = \mathcal{J}(u_1 + u_2) + \alpha_i \|u_i - a\|_2^2 - \|T(u_i - a)\|_2^2, \tag{3.2}
\]

\[
\mathcal{J}_i(u_1, u_2; a) = \mathcal{J}(u_1 + u_2) + \alpha_i \|u_i - a\|_2^2 - \|T(u_i - a)\|_2^2, \tag{3.2}
\]
where $\alpha_1, \alpha_2$ are positive constants chosen as specified below in order to ensure convergence of the subminimization iteration
\[
u_i^{(n+1,\ell+1)} = \arg\min_{u_i \in V_i} \mathcal{J}(u_1, u_2; u_i^{(n+1,\ell)}), \quad \ell > 0,
\]
to a minimizer of the corresponding subproblem of (1.9), i.e.,
\[
\arg\min_{u_i \in V_i} \mathcal{J}(u_1 + u_2),
\]
for $i = 1, 2$. Let us further define the synthesis operators $S_1 : \ell_2 \to V_1$ via the orthonormal basis for $V_1$ and $S_2 : \ell_2 \to V_2$ via the orthonormal basis for $V_2$. That is $u_1 = S_1(u_{\Lambda_1})$ and $u_2 = S_2(u_{\Lambda_2})$ for $u_{\Lambda_1} = (u_\lambda)_{\lambda \in \Lambda_1}$ the scaling function coefficients and $u_{\Lambda_2} = (u_\lambda)_{\lambda \in \Lambda_2}$ the wavelet coefficients. Since $S_1, S_2$ are isometries, we know that
\[
\|T_{V_i}(u_i - a)\|_2^2 = \|T_{V_i}S_i(u_{\Lambda_i} - a_{\Lambda_i})\|_2^2 \quad \text{and} \quad \|u_i - a\|_2^2 = \|u_{\Lambda_i} - a_{\Lambda_i}\|_2^2,
\]
where $a = S_1(a_{\Lambda_1})$ or $a = S_2(a_{\Lambda_2})$ and $T_{V_i}$ denotes the operator $T$ restricted to the subspace $V_i$, for $i = 1, 2$. Because of these observations it makes sense to choose
\[
1 \geq \alpha_i > \|T_{V_i}S_i\|^2
\]
for $i = 1, 2$. Then we obtain
\[
\|T_{V_i}(u^i - a)\|_2^2 = \|T_{V_i}S_i(u_{\Lambda_i} - a_{\Lambda_i})\|_2^2 \leq \|T_{V_i}S_i\|^2 \|u_{\Lambda_i} - a_{\Lambda_i}\|_2^2 < \alpha_i \|u_i - a\|_2^2.
\]
Notice that with constants $\alpha_i$ as in (3.4), we have for $n = 1, 2, \ldots$,
\[
\mathcal{J}(u^{(n)}) \leq \mathcal{J}_2(u_1^{(n,L)}, u_2^{(n,M)}; u_2^{(n-1,M)}) \leq \mathcal{J}(u_1^{(n,L)} + u_2^{(n-1,M)})
\]
\[
\leq \mathcal{J}_1(u_1^{(n,L)}, u_2^{(n-1,M)}; u_2^{(n-1,L)}) \leq \mathcal{J}(u^{(n-1)}).
\]

### 3.2. An alternating minimization.
A simple calculation shows that $\mathcal{J}_i$ can be written in the following form:
\[
\mathcal{J}_i(u_1, u_2; a) = \|T(u_1 + u_2) - g\|_2^2 + 2\alpha_i |\nabla(u_1 + u_2)(\Omega) + \alpha_i\|u_1 - a\|_2^2 - \|T(u_1 - a)\|_2^2 + 2\alpha_i |\nabla(u_1 + u_2)(\Omega) + \phi(a, g, u_i),
\]
where
\[
z_i = \pi_{V_i} a + \frac{1}{\alpha_i} \pi_{V_i}(T^*(g - T(u_i + a)))
\]
and $\phi$ is a function depending only on $a, g, u_i$, and $i \in \{1, 2\} \setminus \{i\}$. Hence,
\[
\arg\min_{u_1 \in V_1} \mathcal{J}_1(u_1, u_2; a) = \arg\min_{u_1 \in V_1} \|u_1 - z_1\|_2^2 + 2\beta_1 |\nabla(u_1 + u_2)(\Omega)|
\]
\[
\arg\min_{u_2 \in V_2} \mathcal{J}_2(u_1, u_2; a) = \arg\min_{u_2 \in V_2} \|u_2 - z_2\|_2^2 + 2\beta_2 |\nabla(u_1 + u_2)(\Omega)|
\]
where $\beta_i = \alpha_i/\alpha_i$, for $i = 1, 2$.

In order to address the subminimization problems (3.6) and (3.7) we have to solve a constrained optimization problem of the type
\[
\arg\min_{\Pi u = 0} \mathcal{J}(u),
\]
where $\Pi$ is a linear bounded operator, specifically an orthogonal projection. More precisely, we have to solve, respectively,
\[
\arg\min_{\pi \pi \Pi u_1 = 0} \mathcal{J}_1(u_1, u_2; a) \quad \text{and} \quad \arg\min_{\pi \pi \Pi u_2 = 0} \mathcal{J}_1(u_1, u_2; a).
\]
There exist a variety of methods that solve this type of constrained minimization problems, as the Augmented Lagrangian Method [27], and its adaptations known under the name of the Bregman iterations [7, 8, 26, 29, 31, 32, 36, 37, 38]. Here, for simplicity, we use the Iterative Oblique Thresholding algorithm as proposed in the work [25]. Before stating the theorem which recalls the iterations [7, 8, 26, 29, 31, 32, 36, 37, 38]. Here, for simplicity, we use the Augmented Lagrangian Method [27], and its adaptations known under the name of the Bregman

There exist a variety of methods that solve this type of constrained minimization problems, as the

**Definition 3.1.** For a convex function $F : \mathcal{H} \rightarrow \mathbb{R}$, we define the subdifferential of $F$ at $u \in \mathcal{H}$, as the set valued function

$$\partial F(u) := \{u^* \in \mathcal{H} : \langle u^*, v-u \rangle + F(u) \leq F(v) \quad \forall v \in \mathcal{H}\}.$$  

It is obvious from this definition that $0 \in \partial F(u)$ if and only if $u$ is a minimizer of $F$. We focus, for instance, on the minimization on $\mathcal{V}_1$, and similar statements hold symmetrically for the minimization on $\mathcal{V}_2$.

**Theorem 3.2.** (Oblique Thresholding, [25]) For $u_2 \in \mathcal{V}_2$ and for $z_1 \in \mathcal{V}_1$ the following statements are equivalent:

(i) $u_1 = \arg \min_{u_1 \in \mathcal{V}_1} \|u_1 - z_1\|^2 + 2\beta_1|\nabla(u_1 + u_2)|(\Omega)$,

(ii) there exists $\eta_1 \in \text{Range}(\pi_{\mathcal{V}_1})^* \simeq \mathcal{V}_2$ such that $0 \in u_1^* - (z_1 - \eta_1) + \beta_1|\nabla(u_1^* + u_2)|(\Omega)$.

(iii) there exists $\eta_1 \in \mathcal{V}_2$ such that $u_1^* = (I - P_{\mathcal{V}_1})(z + u_2 - \eta_1) - u_2 \in \mathcal{V}_1$.

(iv) there exists $\eta_1 \in \mathcal{V}_2$ such that $\eta_1 = \pi_{\mathcal{V}_2} P_{\mathcal{V}_1}(\eta_1 - (z + u_2))$.

The existence of $\eta_1 \in \mathcal{V}_2$ as in the previous theorem is shown in [25, Proposition 4.6]. Moreover, the iteration (3.3) for $i = 1$ can be explicitly rewritten as

$$u_1^{(i+1)} = (I - P_{\mathcal{V}_1})\left(u_1^{(i)} + \frac{1}{\alpha_1}\pi_{\mathcal{V}_1}T^*(g - Tu_2 - Tu_1^{(i)}) + u_2 - \eta_1^{(i)}\right) - u_2,$$

where $\eta_1^{(i)} \in \mathcal{V}_2$ is any solution of the fixed point iteration

$$\eta_1 = \pi_{\mathcal{V}_2} P_{\mathcal{V}_1}\left(\eta_1 - (u_1^{(i)} + \frac{1}{\alpha_1}\pi_{\mathcal{V}_1}T^*(g - Tu_2 - Tu_1^{(i)}) + u_2)\right).$$

The computation of $\eta_1^{(i)}$ can be in fact implemented as the limit of the following fixed point algorithm

$$\eta_1^{(0,\ell)} \in \mathcal{V}_2, \quad \eta_1^{(m+1,\ell)} = \pi_{\mathcal{V}_2} P_{\mathcal{V}_1}\left(\eta_1^{(m,\ell)} - (u_1^{(\ell)} + \frac{1}{\alpha_1}\pi_{\mathcal{V}_1}T^*(g - Tu_2 - Tu_1^{(\ell)}) + u_2)\right), \quad m \geq 0.$$

For the subspace $\mathcal{V}_2$ one can formulate analogous statements just by adjusting the notations accordingly.

Let us return to our sequential algorithm in (1.9) and express it explicitly as follows: pick an initial $\mathcal{V}_1 \oplus \mathcal{V}_2 \ni u_1^{(0,L)} + u_2^{(0,M)} := u^{(0)}$ and iterate for $n = 0, 1, 2, \ldots$,

$$\left\{\begin{array}{ll}
\quad u_1^{(n+1,0)} &= u_1^{(n,L)} \\
\quad u_1^{(n+1,\ell+1)} &= \arg \min_{u_1 \in \mathcal{V}_1} J_1(u_1, u_2^{(n,M)}; u_1^{(n+1,\ell)}) \quad \ell = 0, \ldots, L - 1 \\
\quad u_2^{(n+1,0)} &= u_2^{(n,M)} \\
\quad u_2^{(n+1,m+1)} &= \arg \min_{u_2 \in \mathcal{V}_2} J_2(u^{(n+1,L)}, u_2; u_2^{(n+1,m)}) \quad m = 0, \ldots, M - 1 \\
\quad u^{(n+1)} &= u_1^{(n+1,L)} + u_2^{(n+1,M)}.
\end{array}\right.$$

Note that we prescribe a finite number $L, M \in \mathbb{N}$ of inner iterations for each subspace respectively. Then from (3.8) we obtain sequences $(u_1^{(n,L)})_n, (u_2^{(n,M)})_n$ and $(z_1^{(n,L)})_n, (z_2^{(n,M)})_n$ such that

$$z_1^{(n+1,L)} = u_1^{(n,L)} + \frac{1}{\alpha_1}\pi_{\mathcal{V}_1}(T^*(g - T(u_1^{(n,L)} + u_2^{(n,M)})))$$

and

$$z_2^{(n+1,M)} = u_2^{(n,M)} + \frac{1}{\alpha_2}\pi_{\mathcal{V}_2}(T^*(g - T(u_1^{(n+1,L)} + u_2^{(n,M)}))).$$
Note that
\[ u_1^{(n+1,L)} = \arg \min_{u \in V_1} \| u - z_1^{(n+1,L)} \|_2^2 + 2\beta_1 |\nabla(u + u_2^n)|/\Omega \]
and
\[ u_2^{(n+1,M)} = \arg \min_{u \in V_2} \| u - z_2^{(n+1,M)} \|_2^2 + 2\beta_2 |\nabla(u_1^{(n+1,L)} + u)|/\Omega. \]

4. Main Result. We do not pursue the analysis of the convergence of the algorithm in (3.8), as its proof is exactly the same as in [25, Theorem 5.1]. At this point, we have to emphasize that the convergence of the algorithm in [25] is not the convergence to a minimizer of the original problem of minimizing the functional \( J \) in (3.1), which is a very important, but unfortunately missing property of the algorithm. Therefore, we would like to investigate further equivalent conditions for the limits of the sequences produced by this algorithm to be minimizers of \( J \) that will lead us to a better understanding of the algorithm and hopefully enable us to answer the question of convergence to a minimizer.

**Theorem 4.1.** We collect properties of minimizers of \( J \) and limits of the algorithm in (3.8) in the following statements.

a) Let \( \zeta, u \in \mathcal{H} \). Then \( \zeta \in \partial J(u) \) if and only if there exists \((\xi_0, \xi) \in \mathcal{H} \times \mathcal{H}^2 \) such that
\[ 1. \| \xi \|_2 \leq \alpha, \]
\[ 2. \langle \xi(x), \nabla u(x) \rangle_{\mathbb{R}^2} + \alpha |\nabla u(x)| = 0 \text{ for all } x \in \Omega, \]
\[ 3. T^*\xi_0 - \text{div}(2\xi) + \zeta = 0, \]
\[ 4. -\xi_0 = 2(Tu - g). \]
In particular \( u \) is a minimizer if and only if the conditions 1.-4. hold for \( \zeta = 0 \).

b) Let \((u^{(n)})_n\) be a sequence produced by (3.8). Then for a strongly convergent subsequence of \((u^{(n)} = u_1^{(n,L)} + u_2^{(n,M)})_n\) with limit \( u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)} \), we have
\[ u_1^{(\infty)} = \arg \min_{u \in V_1} \| u - z_1^{(\infty)} \|_2^2 + 2\beta_1 |\nabla(u + u_2^{(\infty)})|/\Omega, \quad (4.1) \]
\[ u_2^{(\infty)} = \arg \min_{u \in V_2} \| u - z_2^{(\infty)} \|_2^2 + 2\beta_2 |\nabla(u_1^{(\infty)} + u)|/\Omega, \quad (4.2) \]
\[ z_1^{(\infty)} = u_1^{(\infty)} + \frac{1}{\alpha_1} \pi_{V_1}(T^*(g - Tu^{(\infty)})), \quad (4.3) \]
\[ z_2^{(\infty)} = u_2^{(\infty)} + \frac{1}{\alpha_2} \pi_{V_2}(T^*(g - Tu^{(\infty)})), \quad (4.4) \]
where \( \beta_i = \alpha/\alpha_i, \) for \( i = 1, 2 \). Moreover let us denote \( z^{(\infty)} = u^{(\infty)} + T^*(g - Tu^{(\infty)}). \) Then, \( u^{(\infty)} \) is a minimizer of (3.1) if and only if
\[ u^{(\infty)} = \arg \min_{u \in \mathcal{H}} \{ F(u) := \| u - z^{(\infty)} \|_2^2 + 2\alpha |\nabla u|/\Omega \}. \quad (4.5) \]

Before proving the previous statements we add some comments on the possibility of verification of the minimality condition (4.5). Let \( F(u_1, u_2) = F(u_1 + u_2) \) for \( u_1 \in V_1 \) and \( u_2 \in V_2 \). Then (4.1) and (4.2) imply
\[ F(u_1^{(\infty)}, u_2^{(\infty)}) \leq \arg \min_{v_1 \in V_1, v_2 \in V_2} \{ F(v_1, u_2^{(\infty)}), F(u_1^{(\infty)}, v_2) \}. \quad (4.6) \]
Unfortunately, (4.6) may not imply that \( u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)} \) is a minimizer of (4.5) and eventually of (3.1). We propose the following univariate counterexample, which also shows that the algorithm in (3.8) may fail to converge to a minimizing solution. For simplicity, we return to the continuous
setting and we assume that $\Omega$ is the interval $[-1,2]$, and $g = \chi_{[0,1/2]}$. We consider univariate Haar wavelets, i.e., let $\varphi_0 = \chi_{[0,1)}$ and $\psi_0 = \chi_{[0,1/2)} - \chi_{[1/2,1)}$. Then we have

$$g = \frac{1}{2} \varphi_0 + \frac{1}{2} \psi_0.$$ 

We can prove the following proposition.

**Proposition 4.2.** Let $0 < \alpha < 1/8$ and $V_1$ be the subspace of $L_2([-1,2])$ generated by $\{\varphi_0(x-k) : k \in \{-1,0,1\}\}$ and $V_2$ be the subspace of $L_2([-1,2])$ generated by $\{\psi_{j,k}(x) = 2^{j/2} \psi_0(2^j x - k) : j \in \mathbb{Z}_+ \cup \{0\}, k \in \{-2^j, \ldots, 2^j\}\}$, then

$$u_1^{(\infty)} = \frac{1 - 4\alpha}{2} \varphi_0, \quad u_2^{(\infty)} = \frac{1 - 4\alpha}{2} \psi_0,$$

which satisfy

$$\arg \min_{u \in V_1} F(u_1, u_2) < F(u_1^{(\infty)}, u_2^{(\infty)}) \leq \arg \min_{v_1 \in V_1, v_2 \in V_2} \left\{ F(v_1, u_2^{(\infty)}), F(u_1^{(\infty)}, v_2) \right\}$$

where

$$F(u_1, u_2) = \|u_1 + u_2 - g\|_2^2 + 2\alpha \|\nabla (u_1 + u_2)\|([-1,2])$$

$$= \left\| u_1 - \frac{1}{2} \varphi_0 \right\|_2^2 + \left\| u_2 - \frac{1}{2} \psi_0 \right\|_2^2 + 2\alpha \|\nabla (u_1 + u_2)\|([-1,2]).$$

**Proof.** We prove the result by showing that the algorithm in (3.8), starting with $u^{(0)} = 0$, stops by converging to $u^{(\infty)} = u_1^{(\infty)} + u_2^{(\infty)}$ in finite iterations, and that (4.7) holds. Let $u_1^{(0)} = u_2^{(0)} = 0$. Then

$$u_1^{(1)} = \arg \min_{u \in V_1} \left\| u - \frac{1}{2} \varphi_0 \right\|_2^2 + 2\alpha \|\nabla u\|([-1,2]).$$

Then $u_1^{(1)} = a \varphi_0$ for some $a > 0$ and

$$\left\| u - \frac{1}{2} \varphi_0 \right\|_2^2 + 2\alpha \|\nabla u\|([-1,2]) = \left\| a \varphi_0 - \frac{1}{2} \varphi_0 \right\|_2^2 + 2\alpha a \|\nabla \varphi_0\|([-1,2])$$

$$= \left( a - \frac{1}{2} \right)^2 + 4\alpha a = \left( a - \frac{4\alpha - 1}{2} \right)^2 + 2\alpha - 4\alpha^2.$$ 

Since $\alpha < 1/8$, (4.8) attains its minimum when

$$a = \frac{1 - 4\alpha}{2},$$

i.e., $u_1^{(1)} = \frac{1 - 4\alpha}{2} \varphi_0$.

Now, we solve

$$u_2^{(1)} = \arg \min_{u \in V_2} \left\| u - \frac{1}{2} \psi_0 \right\|_2^2 + 2\alpha \|\nabla (u_1^{(1)} + u)\|([-1,2]).$$

(4.9)

It is not hard to see that $u_2^{(1)} = b \psi_0$ for some $b > 0$. If we assume $b \leq \frac{1 - 4\alpha}{2}$, then

$$\left\| u - \frac{1}{2} \psi_0 \right\|_2^2 + 2\alpha \|\nabla (u_1^{(1)} + u)\|([-1,2]) = \left( b - \frac{1}{2} \right)^2 + 2\alpha \left( b - \frac{1 - 4\alpha}{2} \right)^2 + 2\alpha - 4\alpha^2.$$

which is minimized when $b = \frac{1 - 4\alpha}{2}$. On the other hand, if we assume $b \geq \frac{1 - 4\alpha}{2}$, then since

$$0 < \frac{1 - 4\alpha}{2} < \frac{1}{2} \leq b,$$

$$\left\| u - \frac{1}{2} \psi_0 \right\|_2^2 + 2\alpha \|\nabla (u_1^{(1)} + u)\|([-1,2]) = \left( b - \frac{1}{2} \right)^2 + 2\alpha \left( b - \frac{1 - 4\alpha}{2} \right)^2 + 2\alpha - 4\alpha^2.$$
It is easy to see that $$u_2^{(1)} = \frac{1 - 4\alpha}{2} \psi_0$$. Hence

$$u_2^{(1)} = \frac{1 - 4\alpha}{2} \psi_0.$$ 

Now, we solve

$$u_1^{(2)} = \arg \min_{u \in V_1} ||u - \frac{1}{2} \varphi_0||_2^2 + 2\alpha |\nabla(u + u_2^{(1)})([-1, 2])|.$$ 

It is easy to see that $$u_1^{(2)} = a \varphi_0$$ for some $$a > 0$$. If we assume $$a \leq \frac{1 - 4\alpha}{2}$$, then since $$\frac{1 - 4\alpha}{2} \leq \frac{1}{2}$$,

$$||u - \frac{1}{2} \varphi_0||_2^2 + 2\alpha |\nabla(u + u_2^{(1)})([-1, 2])| = (a - \frac{1}{2})^2 + 2\alpha (a + \frac{1 - 4\alpha}{2} + (1 - 4\alpha) + \frac{1 - 4\alpha}{2} - a)$$ 

$$= (a - \frac{1}{2})^2 + 4\alpha (1 - 4\alpha) \geq 4\alpha - 12\alpha^2,$$

which is minimized when $$a = \frac{1 - 4\alpha}{2}$$. On the other hand, if we assume $$a \geq \frac{1 - 4\alpha}{2}$$, then

$$||u - \frac{1}{2} \varphi_0||_2^2 + 2\alpha |\nabla(u + u_2^{(1)})([-1, 2])| = (a - \frac{1}{2})^2 + 2\alpha (a + \frac{1 - 4\alpha}{2} + (1 - 4\alpha) + a - \frac{1 - 4\alpha}{2})$$ 

$$= (a + 4\alpha - 1)^2 + 4\alpha - 12\alpha^2 \geq 4\alpha - 12\alpha^2,$$

which is also minimized when $$a = \frac{1 - 4\alpha}{2}$$. We finally obtain

$$u_1^{(2)} = \frac{1 - 4\alpha}{2} \varphi_0 = u_1^{(1)}.$$ 

Therefore, after only one step of the algorithm in (3.8), we have

$$u_1^{(\infty)} = \frac{1 - 4\alpha}{2} \varphi_0, \quad u_2^{(\infty)} = \frac{1 - 4\alpha}{2} \psi_0.$$ 

It is now easy to see that $$u_1^{(\infty)}, u_2^{(\infty)}$$ satisfy (4.6) and

$$F(u_1^{(\infty)}, u_2^{(\infty)}) = 4\alpha - 8\alpha^2.$$ 

However, if $$u = a \chi_{[0, 1/2]} = \frac{a}{2} \varphi_0 + \frac{a}{2} \psi_0$$, then

$$F(u) = ||u - g||_2^2 + 2\alpha |\nabla u|([-1, 2])| = (a - 1)^2 \|\chi_{[0, 1/2]}\|_2^2 + 2\alpha \cdot 2a$$ 

$$= \frac{1}{4} (a - 1)^2 + 4\alpha = \frac{1}{4} (a + (8\alpha - 1))^2 + 4\alpha - 16\alpha^2.$$ 

Since $$0 < \alpha < 1/8$$, if we set $$u_0 = (1 - 8\alpha) \chi_{[0, 1/2]} = \frac{1 - 8\alpha}{2} \varphi_0 + \frac{1 - 8\alpha}{2} \psi_0$$, then

$$\min_{u_1 \in V_1, u_2 \in V_2} F(u_1, u_2) \leq F(u_0) = 4\alpha - 16\alpha^2 < 4\alpha - 8\alpha^2 = F(u_1^{(\infty)} + u_2^{(\infty)}) = F(u_1^{(\infty)}, u_2^{(\infty)}).$$

\[ \square \]

Theorem 4.1-a) also provides us with the following useful characterization.

**Corollary 4.3.** The subdifferential of $$\partial \mathcal{J}(u)(\Omega)$$ is fully characterized by

$$\partial \mathcal{J}(u)(\Omega) = \{ \text{div}(\xi) \in \mathcal{H} : ||\xi||_\infty \leq \alpha, \ (\xi(x), \nabla u(x))_{\mathbb{R}^2} + \alpha |\nabla u|(x) = 0 \text{ for all } x \in \Omega \}$$

$$= \{ \text{div}(\xi) \in \mathcal{H} : -\text{div}(\xi) = P_{\alpha K}(-u - \text{div}(\xi)) \}.$$

**Proof.** If we consider $$T = I$$ in Theorem 4.1-a), then $$\tilde{\zeta} \in \partial \mathcal{J}(u)(\Omega)$$ if and only if $$\zeta = 2(\tilde{\zeta} + u - g) \in \partial \mathcal{J}(u)$$ if and only if there exists $$(\xi_0, \xi) \in \mathcal{H} \times \mathcal{H}^2$$ such that
1. $\|\xi\|_{\infty} \leq \alpha$,
2. $\langle \xi(x), \nabla u(x) \rangle_{\mathbb{R}^2} + \alpha |\nabla u(x)| = 0$ for all $x \in \Omega$,
3. $\zeta = \text{div}(\xi)$.

Hence,

$$\alpha \text{div} |\nabla u|(\Omega) = \{ \text{div}(\xi) \in \mathcal{H} : \|\xi\|_{\infty} \leq \alpha, \langle \xi(x), \nabla u(x) \rangle_{\mathbb{R}^2} + \alpha |\nabla u(x)| = 0 \text{ for all } x \in \Omega \}.$$ 

We also notice that

$$\text{div}(\xi) \in \alpha |\nabla u|(\Omega) \text{ if and only if } 0 \in u - (u + \text{div}(\xi)) + \alpha \text{div} |\nabla u|(\Omega),$$

which is equivalent to

$$u = \arg \min_v \|v - (u + \text{div}(\xi))\|^2 + 2\alpha |\nabla v|(\Omega),$$

that is,

$$-u = \arg \min_v \|v + (u + \text{div}(\xi))\|^2 + 2\alpha |\nabla v|(\Omega).$$

By [25, Examples 4.2.2], the latter optimality problem is equivalent to

$$-u = (I - P_{\alpha K})(-u - \text{div}(\xi)), $$

that is,

$$- \text{div}(\xi) = P_{\alpha K}(-u - \text{div}(\xi)).$$

Therefore, we also have

$$\alpha \text{div} |\nabla u|(\Omega) = \{ \text{div}(\xi) \in \mathcal{H} : -\text{div}(\xi) = P_{\alpha K}(-u - \text{div}(\xi)) \}.$$ 

\[ \square \]

5. Proof of Theorem 4.1.

a) The proof of this statement, which characterizes the minimizers of $J$, can be found in [24, Appendix A].

b) For simplicity, we rename a convergent subsequence again by $(u^{(n)} = u_1^{(n, L)} + u_2^{(n, M)})_n$.

Equations (4.3) and (4.4) follow directly from (3.9) for $n \to \infty$. Furthermore, it is also easy to see that for any $u_1 \in V_1$,

$$\|u_1^{(\infty)} - z_1^{(\infty)}\|^2 + 2\beta_1 |\nabla u^{(\infty)}|(\Omega) = \lim_{n \to \infty} \|u_1^{(n, L)} - z_1^{(n, L)}\|^2 + 2\beta_1 |\nabla (u_1^{(n, L)} + u_2^{(n, M)})|(\Omega)$$

$$\leq \lim_{n \to \infty} \|u_1 - z_1^{(n, L)}\|^2 + 2\beta_1 |\nabla (u_1 + u_2^{(n, M)})|(\Omega)$$

$$= \|u_1 - z_1^{(\infty)}\|^2 + 2\beta_1 |\nabla (u_1 + u_2^{(\infty)})|(\Omega).$$

The second limit is a consequence of [25, formula (5.7)], which states the asymptotic regularity of the sequence, i.e.,

$$\left( \sum_{\ell=0}^{L-1} \|u_1^{(n+1, \ell+1)} - u_1^{(n+1, \ell)}\|^2 + \sum_{m=0}^{M-1} \|u_2^{(n+1, m+1)} - u_2^{(n+1, m)}\|^2 \right) \to 0, \ n \to \infty. \quad (5.1)$$

Hence, we have

$$u_1^{(\infty)} = \arg \min_{u \in V_1} \|u - z_1^{(\infty)}\|^2 + 2\beta_1 |\nabla (u + u_2^{(\infty)})|(\Omega).$$
With the same argument one obtains (4.2). By Theorem 3.2 the optimality conditions (4.1) and (4.2) are equivalent to
\begin{align*}
0 & \in u_1^{(\infty)} - (z_1^{(\infty)} - \eta_1^{(\infty)}) + \beta_1 \partial |\nabla u^{(\infty)}|(\Omega), \\
0 & \in u_2^{(\infty)} - (z_2^{(\infty)} - \eta_2^{(\infty)}) + \beta_2 \partial |\nabla u^{(\infty)}|(\Omega).
\end{align*}
Then by Corollary 4.3 there exist \( \xi_1, \xi_2 \) such that
\begin{align*}
\text{div}(\xi_1) &= -u_1^{(\infty)} + (z_1^{(\infty)} - \eta_1^{(\infty)}), \\
\text{div}(\xi_2) &= -u_2^{(\infty)} + (z_2^{(\infty)} - \eta_2^{(\infty)}),
\end{align*}
and with the following additional properties
1. \( \|\xi\|_\infty \leq \beta_1, \|\xi\|_\infty \leq \beta_2 \) and
2. \( \langle \xi(x), \nabla u^{(\infty)}(x) \rangle_{\mathbb{R}^2} + \beta_j |\nabla u^{(\infty)}(x)| = 0 \) for all \( x \in \Omega \) and \( i = 1, 2 \).

Multiplying (5.2) by \( \alpha_1 \) and (5.3) by \( \alpha_2 \) yields
\begin{align*}
-\alpha_1 u_1^{(\infty)} + \alpha_1 z_1^{(\infty)} - \alpha_1 \eta_1^{(\infty)} - \alpha_1 \text{div}(\xi_1) &= 0, \\
-\alpha_2 u_2^{(\infty)} + \alpha_2 z_2^{(\infty)} - \alpha_2 \eta_2^{(\infty)} - \alpha_2 \text{div}(\xi_2) &= 0.
\end{align*}
If we sum up the last two equations we obtain
\begin{align*}
-\alpha_1 u_1^{(\infty)} + \alpha_1 z_1^{(\infty)} - \alpha_2 u_2^{(\infty)} + \alpha_2 z_2^{(\infty)} - \text{div}(\alpha_1 \xi_1) - \text{div}(\alpha_2 \xi_2) - (\alpha_1 \eta_1^{(\infty)} + \alpha_2 \eta_2^{(\infty)}) &= 0
\end{align*}
From Theorem 3.2 we have that
\begin{align*}
\eta_1^{(\infty)} &= \pi_{V_1} P_{\beta_1 K}(\eta_1^{(\infty)} - (z_1^{(\infty)} + u_2^{(\infty)})), \\
\eta_2^{(\infty)} &= \pi_{V_1} P_{\beta_2 K}(\eta_2^{(\infty)} - (z_2^{(\infty)} + u_1^{(\infty)})�)
\end{align*}
and it follows then from (5.2), (5.3), and Corollary 4.3 that
\begin{align*}
\alpha_1 \eta_1^{(\infty)} &= \pi_{V_1} (\text{div}(\alpha_1 \xi_1)) = \pi_{V_1} P_{\alpha K}(-u^{(\infty)} - \text{div}(\alpha_1 \xi_1)), \\
\alpha_2 \eta_2^{(\infty)} &= \pi_{V_1} (\text{div}(\alpha_2 \xi_2)) = \pi_{V_1} P_{\alpha K}(-u^{(\infty)} - \text{div}(\alpha_2 \xi_2)).
\end{align*}
Plugging (5.5) and (5.6) in (5.4) and using the definition of \( z_1^{(\infty)} \) and \( z_2^{(\infty)} \) yield
\begin{align*}
0 &= -T^* (T u^{(\infty)} - g) - \text{div}(\alpha_1 \xi_1) - \text{div}(\alpha_2 \xi_2) + (\pi_{V_1} \text{div}(\alpha_1 \xi_1) + \pi_{V_2} \text{div}(\alpha_2 \xi_2)) \\
&= -T^* (T u^{(\infty)} - g) - (\pi_{V_1} \text{div}(\alpha_1 \xi_1) + \pi_{V_2} \text{div}(\alpha_2 \xi_2))
\end{align*}
Therefore, if there exists \( \xi \) such that \( \text{div}(\xi) \in \partial |\nabla u^{(\infty)}|(\Omega) \) and
\begin{align*}
\text{div}(\xi) &= \pi_{V_1} \text{div}(\alpha_1 \xi_1) + \pi_{V_2} \text{div}(\alpha_2 \xi_2),
\end{align*}
then \( \xi \) also satisfies
1. \( \|\xi\|_\infty \leq \alpha \),
2. \( \langle \xi(x), \nabla u^{(\infty)}(x) \rangle_{\mathbb{R}^2} + \alpha |\nabla u^{(\infty)}|(x) = 0 \) for all \( x \in \Omega \),
3. \( T^* \xi_0 - \text{div}(2\xi) = 0 \),
4. \( -\xi_0 = 2(T u^{(\infty)} - g) \).
The existence of such \( \xi \) is a necessary and sufficient condition for \( u^{(\infty)} \) to be a minimizer by a). Then \( \xi \) satisfies
\begin{align*}
-u^{(\infty)} + z^{(\infty)} &= T^* (g - T u^{(\infty)}) = -\alpha_1 u_1^{(\infty)} + \alpha_1 z_1^{(\infty)} - \alpha_2 u_2^{(\infty)} + \alpha_2 z_2^{(\infty)} = \text{div}(\xi),
\end{align*}
that is,
\begin{align*}
-z^{(\infty)} &= -u^{(\infty)} - \text{div}(\xi) \quad \text{and} \quad \text{div}(\xi) \in \partial |\nabla u^{(\infty)}|(\Omega),
\end{align*}
where \( z^{(\infty)} := u^{(\infty)} + T^*(g - Tu^{(\infty)}) \). Note that for \( i = 1, 2 \),

\[
\pi_{\nu_i} z^{(\infty)} = (1 - \alpha_i)u^{(\infty)} + \alpha_i z_i^{(\infty)}.
\]

By \( \text{div}(\xi) \in \alpha \partial |\nabla u^{(\infty)}|(\Omega) \) and Corollary 4.3, this is equivalent to

\[
u^{(\infty)} - z^{(\infty)} = - \text{div}(\xi) = P_{\alpha K}(-u^{(\infty)} - \text{div}(\xi)) = P_{\alpha K}(-z^{(\infty)})
\]

Hence,

\[-u^{(\infty)} = (I - P_{\alpha K})(z^{(\infty)}) = \arg \min_u \| u + z^{(\infty)} \|^2 + 2\alpha |\nabla u|(\Omega).
\]

which proves the theorem.

The proof of Theorem 4.1 finally provides us with another characterization of \( u^{(\infty)} \) being a minimizer of (3.1) by \( \xi_1, \xi_2 \) in (5.2), (5.3).

**Corollary 5.1.** Let \( \alpha_1 \leq 1, \alpha_2 \leq 1 \). The limit \( u^{(\infty)} \), obtained in Theorem 4.1 b), is a minimizer of (3.1) if and only if there exist \( \xi_1, \xi_2 \) in (5.2), (5.3) with \( \text{div}(\alpha_1 \xi_1) = \text{div}(\alpha_2 \xi_2) \).

**Proof.** First let us prove the statement for \( \alpha_1 = \alpha_2 = 1 \): If \( u^{(\infty)} \) is a minimizer of (3.1), then Theorem 4.1 and [25, Examples 4.2.2] say that

\[ u^{(\infty)} = (I - P_{\alpha K})(z^{(\infty)}). \]

Since \( \alpha_1 = \alpha_2 = 1 \), we obtain

\[ z_1^{(\infty)} = \pi_{\nu_1} z^{(\infty)}, \quad z_2^{(\infty)} = \pi_{\nu_2} z^{(\infty)}. \]

We then can rephrase this in two different ways as follows.

\[
u_1^{(\infty)} = (I - P_{\alpha K})(z_1^{(\infty)} + u_2^{(\infty)} - (u_2^{(\infty)} - z_2^{(\infty)})) - u_2^{(\infty)},
\]

or

\[
u_2^{(\infty)} = (I - P_{\alpha K})(z_2^{(\infty)} + u_1^{(\infty)} - (u_1^{(\infty)} - z_1^{(\infty)})) - u_1^{(\infty)}.
\]

By Theorem 3.2, we can take

\[ \eta_1^{(\infty)} = u_2^{(\infty)} - z_2^{(\infty)}, \quad \eta_2^{(\infty)} = u_1^{(\infty)} - z_1^{(\infty)}. \]

This implies \( \text{div}(\xi_1) = \text{div}(\xi_2) \) from (5.2) and (5.3). On the other hand, if \( \text{div}(\xi_1) = \text{div}(\xi_2) \), then (5.7) implies that \( u^{(\infty)} \) is a minimizer of (3.1).

Now let us prove the statement for \( \alpha_1, \alpha_2 \leq 1 \): Suppose that \( u^{(\infty)} \) is a minimizer of (3.1). Then Theorem 4.1 b) says that

\[ u^{(\infty)} = (I - P_{\alpha K})(z^{(\infty)}) \] if and only if \( \text{div}(\xi) = -u^{(\infty)} + z^{(\infty)} \in \alpha \partial |\nabla u^{(\infty)}|(\Omega) \) for some \( \xi \).

By the above considerations, we know that there exist \( \eta_1^{(\infty), 1}, \eta_2^{(\infty), 1} \) such that

\[
\eta_1^{(\infty), 1} = u_2^{(\infty)} - \pi_{\nu_1} z^{(\infty)} = \alpha_2 u_2^{(\infty)} - \alpha_2 z_2^{(\infty)}, \quad \eta_2^{(\infty), 1} = u_1^{(\infty)} - \pi_{\nu_2} z^{(\infty)} = \alpha_1 u_1^{(\infty)} - \alpha_1 z_1^{(\infty)}.
\]

and

\[
-u_1^{(\infty)} + (\pi_{\nu_1} z^{(\infty)} - \eta_1^{(\infty), 1}) = \text{div}(\xi) = -u_2^{(\infty)} + (\pi_{\nu_2} z^{(\infty)} - \eta_2^{(\infty), 1}).
\]

Let \( \eta_1^{(\infty), \alpha_i} = \frac{\eta_1^{(\infty), 1}}{\alpha_i}, \xi_1^{(\infty)} = \frac{\xi_1}{\alpha_i} \) for \( i = 1, 2 \). Then

\[
\text{div}(\xi_1^{(\infty)}) = -u_1^{(\infty)} + (z_1^{(\infty)} - \eta_1^{(\infty), \alpha_1}),
\]

\[
\text{div}(\xi_2^{(\infty)}) = -u_2^{(\infty)} + (z_2^{(\infty)} - \eta_2^{(\infty), \alpha_2}).
\]

Moreover one can see that \( \text{div}(\xi_i^{(\infty)}) \in \beta \partial |\nabla u^{(\infty)}|(\Omega) \) for \( i = 1, 2 \). Hence if we let \( \xi_1 = \xi_1^{(\infty)} \) and \( \xi_2 = \xi_2^{(\infty)} \), then \( \text{div}(\alpha_1 \xi_1) = \text{div}(\xi) = \text{div}(\alpha_2 \xi_2) \).

On the other hand, if there exist \( \xi_1, \xi_2 \) satisfying \( \text{div}(\alpha_1 \xi_1) = \text{div}(\alpha_2 \xi_2) \) in (5.2), (5.3), then by (5.7), we know that the limit \( u^{(\infty)} \) is a minimizer of (3.1).
6. Numerical Validation. In this section we illustrate the performance of the algorithm in (3.8) for the minimization of (3.1) when \( T \) is a blur operator with averaging kernel \( \kappa \). The function space is split into \( N \in \mathbb{N} \) orthogonal spaces by a wavelet space decomposition such that

\[
\mathcal{H} = V_{2-N} \oplus \left( \bigoplus_{j=2-N}^{0} W_j \right)
\]

and we set \( V_1 := V_{2-N} \) and \( V_i := W_{2-i} \) for \( i = 2, 3, \ldots, N \). Note that for \( N = 1 \) we have that \( V_1 = \mathcal{H} \) and thus we have no splitting. In order to gain maximal performance, the preconditioner constants are always chosen as

\[
\alpha_i > \| T V_i S_i \|^2,
\]

for \( i = 1, \ldots, N \), as already discussed in detail for \( N = 2 \) in Section 3.1. In the case \( N = 1 \), where no decomposition is done, we just set the preconditioner constant \( \alpha_1 = 1 \), which is a good choice in order to verify how big the preconditioning impact for \( N > 1 \) for deblurring problems is. However, in order to ensure convergence the assumption \( \| T \| < 1 \) has to be fulfilled, which might not be true for a general blurring operator \( T \). Therefore, we simply rescale the minimization problem by multiplying the functional (3.1) by the positive constant \( \frac{1}{\| T \|^2 + \varepsilon} \), for \( \varepsilon > 0 \) fixed. Let us emphasize that such a rescaling does not change the minimizer of the functional at all. However, we are aware of the fact that rescaling in a “good” way is already preconditioning the problem, see Table 6.2. Note that a rescaling is redundant when preconditioner constants are chosen as in (6.1), since this choice already ensures convergence of the algorithm, see Section 3. Moreover any rescaling of the whole problem automatically effects the preconditioner constant in an equivalent way, i.e., if we rescale the functional \( \mathcal{J} \) in (3.1) by \( \gamma > 0 \), then the rescaled preconditioner constants \( \alpha_i^\gamma \) are chosen according to (6.1) as

\[
\alpha_i^\gamma = \gamma \alpha_i > \gamma \| T V_i S_i \|^2 = \| \sqrt{\gamma} T V_i S_i \|^2,
\]

which is then equivalent to just multiplying (3.2) by \( \gamma \).

In our numerical examples we only consider decompositions by using Haar wavelets. In this case it is easy to see that the preconditioner constant for the scale space \( V_{2-N} \) is simply \( \alpha_1 > \| T \| \) and the preconditioner constants for the wavelet spaces \( W_j, j = 0, \ldots, 2-N \), are strictly smaller than the norm of \( T \).

The implementation of the algorithm is done as suggested and discussed in [25]. That is the subiterations in (3.8) are solved by computing the minimizers by means of oblique thresholding, cf. Theorem 3.2. For the computation of the orthogonal projection onto \( \beta_i K, i = 1, \ldots, N \), in the oblique thresholding we use an algorithm proposed by Chambolle in [9].

6.1. Experiments. As a measure for the restoration quality of an image we would like to use a “distance” between the obtained estimate and the original image. Therefore we recall the definition of Signal-to-Error-Ratio Gain [35] given by

\[
\text{SERG} = 20 \log_{10} \frac{\| g - \text{org} \|}{\| u^* - \text{org} \|},
\]

where \( \text{org} \) denotes the original image before blurring and \( u^* \) the restored image. In our examples we stop the algorithm in (3.8) as soon as the quality measure \( \text{SERG} \) reaches a significant level \( S^* \), i.e.,

\[
\text{SERG} \geq S^*. \tag{6.2}
\]

We start our numerical discussion in Figure 6.1, where we show an image of size 156 × 156 pixels, which is corrupted by an averaging blur operator \( T \) with kernel \( \kappa \) supported on 9 × 9 pixels and uniform values 1/81. In order to deblur this image we split the function space of the image into orthogonal subspaces via a wavelet space decomposition and compute its solution by the
algorithm with \( \alpha = \frac{2}{3} \cdot 10^{-4} \) and stopping criterion (6.2) with \( S^* = 3.3 \). Here, the regularisation parameter \( \alpha \) is chosen according to the strength of the blurring, i.e., the value of \( \alpha \) increases with increasing size of the blurring kernel. At the same time, the SERG value \( S^* \) we can hope to reach decreases with increasing the blur on the image. The computed result for 5 subspaces is shown in Figure 6.1 on the right hand side.

![Blurred and restored images](image)

**Fig. 6.1.** On the left we depict an image, blurred with an averaging kernel of size 9 \times 9 pixels with uniform values \( 1/81 \). On the right we show the corresponding solution computed on 5 orthogonal subspaces by the algorithm in (3.8) with \( \alpha = \frac{2}{3} \cdot 10^{-4} \) and stopping criterion (6.2) with \( S^* = 3.3 \).

With the same setting as above we solve this specific deblurring problem with the algorithm in (3.8) for different numbers of subspaces and compare its performance with respect to the needed iterations and computational time in Table 6.1. Note that for \( N = 1 \) we solve this problem without any decomposition on the whole space \( \mathcal{H} \). We see in Table 6.1 that the performance in this case is the worst. When we solve the same problem with a decomposition into two or more wavelet spaces fewer iterations are needed to reach the stopping criterion, which is due to the preconditioning. This also leads to a speed-up in computational time that is most significant for a splitting into 5 subspaces.

<table>
<thead>
<tr>
<th>( N )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>832</td>
<td>323</td>
<td>148</td>
<td>84</td>
<td>53</td>
<td>54</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>88.98</td>
<td>76.80</td>
<td>55.49</td>
<td>40.64</td>
<td>32.26</td>
<td>41.04</td>
</tr>
</tbody>
</table>

**Table 6.1**

Performance of the wavelet decomposition algorithm in (3.8) for the image deblurring problem of Figure 6.1 (Average kernel 9 \times 9 with uniform value \( 1/81 \), image size 156 \times 156 pixels) with \( \alpha = \frac{2}{3} \cdot 10^{-4} \) and SERG-stopping criterion (6.2) with \( S^* = 3.3 \): the number of iterations and CPU time in seconds are shown with respect to the number \( N \) of subspace decompositions.

Note that since the norm of \( T \) might exceed 1, we rescale the problem for \( N = 1 \) with \( \frac{1}{\|T\|+\varepsilon} \), \( \varepsilon > 0 \) such that we can guarantee convergence. For simplicity we used \( \varepsilon = 1 \) to obtain a reasonable rescaling. We also tested the algorithm with different (smaller) \( \varepsilon \)-values, see Table 6.2. Note that rescaling the problem with a very small \( \varepsilon > 0 \) already introduces a preconditioning effect in the problem as we clearly see in Table 6.2. Nevertheless, with our orthogonal wavelet decomposition strategy into 5 subspaces the algorithm in (3.8) still performs clearly better.

We further tested our wavelet algorithm for varying sizes of averaging kernels and for Gaussian blur. In particular, we consider small averaging kernels of size 3 \times 3 pixels with values 1/9, see Figure 6.2, and large averaging kernels of size 11 \times 11 pixels with values 1/121, see Figure 6.3. As before, these specific deblurring problems were solved with the algorithm in (3.8) for different numbers of subspaces and we compare their performances with respect to the needed iterations and computational time in Table 6.3 and Table 6.4. Note that for \( N = 1 \), where no decomposition
Performance of the wavelet decomposition algorithm in (3.8) for the image deblurring problem of Figure 6.1 (Average kernel $9 \times 9$ with uniform value $\frac{1}{81}$, image size $156 \times 156$ pixels) with SERG-stopping criterion (6.2) with $S^* = 3.3$. Since it might be that $\|T\| > 1$, we have to rescale the problem for $N = 1$ with $\frac{1}{\|T\| + \epsilon}$ in order to guarantee convergence: the number of iterations and CPU time in seconds are shown with respect to the number $N$ of subspace decompositions with different $\epsilon$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\epsilon$</th>
<th>Iterations</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>832</td>
<td>88.98</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>624</td>
<td>61.73</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>457</td>
<td>49.13</td>
</tr>
<tr>
<td>1</td>
<td>$10^{-5}$</td>
<td>415</td>
<td>43.25</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>53</td>
<td>32.26</td>
</tr>
</tbody>
</table>

Table 6.2

is done, we rescale the problem here and in the sequel always as above by $\frac{1}{\|T\|}$ with $\epsilon = 1$. In the case of a small blurring kernel of size $3 \times 3$ the advantage of the wavelet decomposition algorithm (3.8) over the deblurring algorithm without decomposition ($N = 1$) almost disappears (a slightly faster CPU time can be achieved with $N = 2$ and $N = 4$). In contrast, when we have a lot of blurring, see Figure 6.3, where a blurring kernel of size $11 \times 11$ was used, a wavelet decomposition in several subspaces clearly improves the computational performance of the deblurring algorithm, see Table 6.4. Hence, the wavelet decomposition strategy pays off in terms of computational time for large blurring kernels, which is usually the interesting case in practice.

In Figure 6.4 we show an image that is blurred by a blurring operator with Gaussian kernel of size $9 \times 9$ pixels and with variance 20. In order to deblur the image we split the function space of the image into 5 orthogonal wavelet spaces and compute its solution by the algorithm in (3.8)
Fig. 6.3. On the left we show an image of $156 \times 156$ pixels, blurred by an averaging kernel supported on $11 \times 11$ pixels with uniform values $1/121$. On the right we show the corresponding solution computed alternating on 5 orthogonal subspaces by the algorithm in (3.8) with $\alpha = \frac{5}{3} \cdot 10^{-4}$ and stopping criterion (6.2) with $S^* = 2.5$.

<table>
<thead>
<tr>
<th>N</th>
<th>Iterations</th>
<th>CPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>817</td>
<td>96.67</td>
</tr>
<tr>
<td>2</td>
<td>390</td>
<td>102.70</td>
</tr>
<tr>
<td>3</td>
<td>224</td>
<td>94.19</td>
</tr>
<tr>
<td>4</td>
<td>133</td>
<td>77.94</td>
</tr>
<tr>
<td>5</td>
<td>67</td>
<td>52.00</td>
</tr>
<tr>
<td>6</td>
<td>69</td>
<td>64.08</td>
</tr>
</tbody>
</table>

Table 6.4: Performance of the wavelet decomposition algorithm in (3.8) for the image deblurring problem of Figure 6.3 (Average kernel $11 \times 11$ with uniform value $1/121$, image size $156 \times 156$ pixels) with $\alpha = \frac{5}{3} \cdot 10^{-4}$ and SERG-stopping criterion (6.2) with $S^* = 2.5$: the number of iterations and CPU time in seconds are shown with respect to the number $N$ of subspace decompositions.

with $\alpha = \frac{5}{3} \cdot 10^{-4}$ and stopping criterion (6.2) with $S^* = 3.3$, see Figure 6.4 right hand side. With the same setting we solve this specific problem for different numbers of subspaces and compare its performance with respect to the needed iterations and computational time in Table 6.5. Also in this example we see a significant improvement when using the wavelet decomposition algorithm.

By using Lemma 5.1 we check for a splitting into 2 orthogonal subspaces whether the sequential
algorithm numerically converges to a minimizer by looking at
\[ \| \text{div}(\alpha_1 \xi_1^{(n)}) - \text{div}(\alpha_2 \xi_2^{(n)}) \|, \tag{6.3} \]
where
\[ \text{div}(\xi_1^{(n)}) = -u_1^{(n)} + (z_1^{(n)} - \eta_1^{(n)}) \]
\[ \text{div}(\xi_2^{(n)}) = -u_2^{(n)} + (z_2^{(n)} - \eta_2^{(n)}). \]

In Figure 6.5 we plot the decay of this norm discrepancy, indicator of the distance from convergence to a minimizer, with respect to the iterations \( n \). The indicator numerically converges to zero for \( n \) increasing and the algorithm numerically converges to a minimizer of the original problem.

![Graph of norm discrepancy](image)

**Fig. 6.5.** We plot \( \| \text{div}(\alpha_1 \xi_1^{(n)}) - \text{div}(\alpha_2 \xi_2^{(n)}) \| \) for the problem of Figure 6.1 (left) and Figure 6.6 (right) in view of Lemma 5.1 in order to check whether the algorithm is indeed converging.

In Figure 6.6 we depict another example of an image deblurring problem, where the image of size 279 × 285 pixels was blurred by a Gaussian kernel with size 11 × 11 pixels and variance 20. The image is again recovered via the algorithm in (3.8) by splitting the function space \( \mathcal{H} \) into orthogonal wavelet spaces. We take as a stopping criterion (6.2) with \( S^* = 5.5 \) and as a regularization parameter \( \alpha = \frac{5}{3} \cdot 10^{-4} \). Note that in this example the image size is bigger than in our first example and the distribution of structural scales in the image in Figure 6.6 is very different to the one in the Barbara image. In particular, the image in Figure 6.6 is made up of finer scale features, like the fur of the cat, while in the Barbara image from before the finest scales are the ones inside the scarf. We will see that both of these discrepancies will cause our algorithm to behave slightly different to its behaviour in the first deblurring example. In Table 6.6 we show the behaviour of the algorithm for different numbers of subspaces. While the preconditioning effect of the decomposition approach in (3.8) still results in a decrease in the number of iterations, it does not improve the performance with respect to the computational time. We think that the reason for this is a combination of the two new aspects mentioned before. Because the image is quite different from the image used in our previous example the preconditioning effect seems to

<table>
<thead>
<tr>
<th>( N )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>895</td>
<td>352</td>
<td>160</td>
<td>90</td>
<td>57</td>
<td>58</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>102.43</td>
<td>83.65</td>
<td>55.65</td>
<td>43.87</td>
<td>35.48</td>
<td>43.74</td>
</tr>
</tbody>
</table>

**Table 6.5**

Performance of the wavelet decomposition algorithm in (3.8) for the image deblurring problem of Figure 6.4 (Gaussian kernel 9 × 9, image size 156 × 156 pixels) with \( \alpha = \frac{5}{3} \cdot 10^{-4} \) and SERG-stopping criterion (6.2) with \( S^* = 3.3 \): the number of iterations and CPU time in seconds are shown with respect to the number \( N \) of subspace decompositions.
be weaker, not decreasing the number of iterations as much as in the previous example. Therefore and because of the almost doubled size of the image, the additional computational effort paid for solving the deblurring problem, e.g., twice (for \( N = 2 \)), instead of once (\( N = 1 \)) is not compensated by the decrease in the number of iterations. In contrast, when we only consider a small piece of the image (e.g., \( 64 \times 64 \) pixel) than the decomposition algorithm in (3.8) performs again better than without decomposition, see Table 6.6. This leads to the suggestion that if a bigger image has to be deblurred one may combine algorithm (3.8) with a parallel domain decomposition strategy, e.g., solving the deblurring problem on subdomains of the image parallel to each other, cf. [25].

![Blurred image](image1.jpg) ![Restored image](image2.jpg)

Fig. 6.6. On the left we show an image, blurred by a Gaussian kernel. On the right we show the corresponding solution computed alternating on 2 orthogonal subspaces by the algorithm in (3.8) with \( \alpha = \frac{5}{3} \cdot 10^{-4} \) and stopping criterion (6.2) with \( S^* = 5.5 \).

<table>
<thead>
<tr>
<th>Image size 279 × 285 and ( S^* = 5.5 )</th>
<th>N</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>612</td>
<td>517</td>
<td>413</td>
<td>201</td>
<td>99</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>CPU (s)</td>
<td>206.92</td>
<td>535.33</td>
<td>723.65</td>
<td>469.81</td>
<td>292.83</td>
<td>355.38</td>
<td></td>
</tr>
<tr>
<td>Image size 156 × 156 and ( S^* = 5.5 )</td>
<td>Iterations</td>
<td>492</td>
<td>318</td>
<td>241</td>
<td>138</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>59.5</td>
<td>87.85</td>
<td>110.56</td>
<td>91.17</td>
<td>58.43</td>
<td>70.17</td>
<td></td>
</tr>
<tr>
<td>Image size 64 × 64 and ( S^* = 5.5 )</td>
<td>Iterations</td>
<td>189</td>
<td>111</td>
<td>72</td>
<td>34</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>9.12</td>
<td>10.66</td>
<td>11.53</td>
<td>8.65</td>
<td>7.30</td>
<td>9.05</td>
<td></td>
</tr>
<tr>
<td>Image size 64 × 64 and ( S^* = 6.5 )</td>
<td>Iterations</td>
<td>314</td>
<td>176</td>
<td>120</td>
<td>51</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>CPU (s)</td>
<td>15.62</td>
<td>17.18</td>
<td>18.28</td>
<td>11.68</td>
<td>9.54</td>
<td>12.34</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.6

Performance of the wavelet decomposition algorithm for image deblurring (Gaussian kernel 11 × 11) with \( \alpha = \frac{5}{3} \cdot 10^{-4} \) and with SERG-stopping criterion (6.2): the number of iterations and CPU time in seconds are shown with respect to the number \( N \) of subspace decompositions.

In Figure 6.8 we show the evolution of the quality measure SERG with respect to time for the deblurring problem in Figure 6.1 for \( N = 1 \) (no splitting), for \( N = 5 \) (splitting into 5 Wavelet subspaces), and for the domain decomposition algorithm of [25] with 2 subdomains. The Wavelet decomposition algorithm in (3.8) achieves the target SERG-value much faster than the other two minimization strategies, which is due to the preconditioning that leads to a tremendous iteration reduction. Note that domain decomposition strategies only really pay off when we deal with large image sizes, as it is not the case in the present example (image size 156 × 156 pixels). However the domain decomposition algorithm of [25] still outperforms the algorithm without decomposition. While the wavelet decomposition algorithm proves to be the most effective one, reaching higher SERG values very fast, all three algorithms decrease the energy \( \mathcal{J} \) nearly with the same speed as we depict in Figure 6.7.
Fig. 6.7. We show for $N = 1$, $N = 5$, and for the domain decomposition algorithm [25] for a decomposition into 2 subdomains the decay of the energy for the deblurring problem of Figure 6.1. In (b) we zoomed in to the first 10 seconds.

Fig. 6.8. We show for $N = 1$, $N = 5$ of the algorithm in (3.8), and for the domain decomposition algorithm [25] for a decomposition into 2 subdomains the evolution of the quality measure SERG for the deblurring problem of Figure 6.1.

6.2. Comparison with other wavelet-based algorithms. In this section we compare the performance of the algorithm in (3.8) with the algorithm of Vonesch and Unser [35] and with the wavelet frame algorithm developed by Shen et al.[6]. For our comparison we choose a deblurring and denoising problem for a fairly simple test image. Here, in Figure 6.9 (a) we show an image of size $128 \times 128$, which is black on the left half and white on the right half. We corrupt this image by a blur operator with averaging kernel supported on $3 \times 3$ pixels and uniform values $1/9$, and we add additive white Gaussian noise with variance 20, see Figure (6.9) (b). We check the performance of all three algorithms concerning the recovery of the edge in the picture. In order to guarantee a fair comparison the deblurring & denoising problem is solved by each of the three algorithms with different parameters. Then, the best restoration of each algorithm in terms of SERG is chosen. For the algorithm in (3.8) we decomposed the problem into two orthogonal wavelet spaces, used the stopping criterion

$$\|u^{(n)} - u^{(n-1)}\| < 10^{-5}$$
and tested for $\alpha = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4},$ and $10^{-5}$. It turned out that the best result was obtained for $\alpha = 10^{-2}$, see Figure 6.10 (a). The algorithm by Vonesch and Unser was tested for regularization parameters $\alpha$ of the value $1, 10, 100, 1000$ and terminated after $20, 50, 100, 500,$ and $1000$ iterations. The best result was obtained for $\alpha = 100$ by 50 iterations, see Figure 6.10 (c). The frame based algorithm by Shen et al. was tested for regularization parameters $\lambda$ of the value $0.05, 0.15, 0.2, 0.3, 0.5$ and $\mu = 0.002, 0.02, 0.2,$ cf. [6] for more details. The best result was obtained for $\mu = 0.2, \lambda = 0, 15$ by 1000 iterations, see Figure 6.10 (e). In order to show more clearly the differences in the reconstruction of these three algorithms, we plotted in the middle column of Figure 6.10 a corresponding cross section and in the right column a zoom of a part of the edge, where we changed the colors (dark blue represents black and dark red represents white) to visualize the edge reconstruction better. Then we see that the algorithm in (3.8) restores the edge between black and white nearly perfectly while the black and white parts stay very uniform, see Figure 6.10 (b) and (c). In contrast the algorithm of Vonesch & Unser has problems in restoring the edge as well as restoring the uniform parts, see Figure 6.10 (e) and (f). The algorithm by Shen et al. also does some slight smoothing around the edge as well as it does some smoothing on the boundaries, see Figure 6.10 (h) and (i). To sum up the algorithm in (3.8) gives the best restoration of this deblurring & denoising problem, i.e., the sharpest reconstruction of the edge. However we observe that the other two algorithms perform there reconstruction much faster although not so accurate.

![Original Image](a)

![Corrupted Image](b)

Fig. 6.9. (a) Original image that is black on its left and white on its right side. (b) Image corrupted by a blur operator with averaging kernel supported on $3 \times 3$ pixels and uniform values $1/9$ and additive white Gaussian noise with variance $20$.

Acknowledgments. Massimo Fornasier, Yunho Kim, and Andreas Langer acknowledge the financial support provided by the FWF project Y 432-N15 START-Preis Sparse Approximation and Optimization in High Dimensions. Carola-B. Schönlieb acknowledges the financial support provided by the DFG Graduiertenkolleg 1023 Identification in Mathematical Models: Synergy of Stochastic and Numerical Methods, the Wissenschaftskolleg (Graduiertenkolleg, Ph.D. program) of the Faculty for Mathematics at the University of Vienna (funded by the Austrian Science Fund FWF) and the FFG project no. 813610 Erarbeitung neuer Algorithmen zum Image Inpainting. Further, this publication is based on work supported by Award No. KUK-II-007-43 , made by King Abdullah University of Science and Technology (KAUST). The results of the paper also contribute to the project WWTF Five senses-Call 2006, Mathematical Methods for Image Analysis and Processing in the Visual Arts.

REFERENCES

Fig. 6.10. Restoration of the corrupted image in Figure 6.9 with (a) the algorithm in (3.8) with a plot of its cross section in (b); (d) the algorithm by Vonesch and Unser [35] with cross section in (e); (g) the algorithm by Shen et al. [6] with a cross section in (h). In the right column we depict a zoom of the edge where the colors are changed such that dark blue is representing black and dark red is representing white, in order to visualize the edge reconstruction better.


