OPEN ACCESS
IOP Publishing Inverse Problems
Inverse Problems **36** (2020) 124004 (33pp) https://doi.org/10.1088/1361-6420/abb299

An inner–outer iterative method for edge preservation in image restoration and reconstruction[*]

Silvia Gazzola[1,], Misha E Kilmer[2], James G Nagy[3], Oguz Semerci[4] and Eric L Miller[5]**

[1] Department of Mathematical Sciences, University of Bath, Bath, United Kingdom
[2] Department of Mathematics, Tufts University, Medford, MA 02155, Unites States of America
[3] Department of Mathematics, Emory University, Atlanta, GA
[4] Spotify New York, Unites States of America
[5] Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155, Unites States of America

E-mail: S.Gazzola@bath.ac.uk, misha.kilmer@tufts.edu, jnagy@emory.edu, oguz.semerci@gmail.com and eric.miller@tufts.edu

Received 31 December 2019, revised 1 August 2020
Accepted for publication 26 August 2020
Published 3 December 2020

Abstract

We present a new inner–outer iterative algorithm for edge enhancement in imaging problems. At each outer iteration, we formulate a Tikhonov-regularized problem where the penalization is expressed in the two-norm and involves a regularization operator designed to improve edge resolution as the outer iterations progress, through an adaptive process. An efficient hybrid regularization method is used to project the Tikhonov-regularized problem onto

[*] SG's effort for this paper is supported in part by the EPSRC under Grants EP/P005985/1 and EP/T001593/1, and the NSF under Grant DMS-1906664. JGN's effort for this paper is supported in part by the US National Science Foundation under Grant DMS-1819042 and the NIH under Grant 1R13EB028700-01. ELM's effort for this paper is based upon work supported by the US Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 2013-ST-061-ED0001 as well as NSF grants 1934553, 1935555, and 1720291. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the US Department of Homeland Security.

[**] Author to whom any correspondence should be addressed.

approximation subspaces of increasing dimensions (inner iterations), while conveniently choosing the regularization parameter (by applying well-known techniques, such as the discrepancy principle or the $\mathcal{L}$-curve criterion, to the projected problem). This procedure results in an automated algorithm for edge recovery that does not involve regularization parameter tuning by the user, nor repeated calls to sophisticated optimization algorithms, and is therefore particularly attractive from a computational point of view. A key to the success of the new algorithm is the design of the regularization operator through the use of an adaptive diagonal weighting matrix that effectively enforces smoothness only where needed. We demonstrate the value of our approach on applications in x-ray CT image reconstruction and in image deblurring, and show that it can be computationally much more attractive than other well-known strategies for edge preservation, while providing solutions of greater or equal quality.

Keywords: edge enhancement, hybrid regularization methods, Krylov subspace methods, parameter choice strategies, iterative reweighting, tomography, image deblurring

(Some figures may appear in colour only in the online journal)

1. Introduction

In this paper, we consider a new inner–outer iterative algorithm for edge enhancement in image restoration and reconstruction problems. Specifically, we consider generating regularized solutions x_{reg} to

$$Ax = b_{\text{true}} + \eta = b, \quad \text{with} \quad b_{\text{true}} := Ax_{\text{true}}, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times n}$ is a known ill-posed forward operator, b represents the known measured data corrupted by some unknown noise η, and x_{true} is the unknown, vectorized version of the true image to which we would like to generate an approximation. Here and in the following we assume that η is Gaussian white noise.

One of the most well-known regularization methods is Tikhonov regularization, which consists in computing

$$x_{\text{reg}} = \arg\min_{x} \|Ax - b\|_2^2 + \lambda^2 R(x), \tag{1.2}$$

where $R(x)$ is referred to as the regularization term. The regularization parameter λ determines the trade-off between the fidelity to the model, and the damping of the noise through enforcement of *a priori* information via R. Indeed, the role of λ cannot be overemphasized, since the quality of the solution estimate is highly dependent on its value. Unfortunately, in practice, good values of λ are rarely known *a priori*. When $R(x)$ is a non-quadratic term, the most widely used approach is to solve (1.2) with a suitable optimization method for many values of λ, and use a heuristic approach, based on the output, to decide which provides the best quality solution. For some choices of $R(x)$ that try to preserve edges in x, this can be a very expensive strategy, as we discuss below.

To motivate the work in this paper, let us first consider the case $R(x) = \|Lx\|_2^2$, where L is taken to be the identity I (standard form Tikhonov) or a discrete derivative-type operator (general form Tikhonov), such as the discrete gradient or Laplacian. Assuming that an appropriate value of λ is known, the corresponding solution x_{reg} will necessarily be smooth given these choices of L, as the magnitude of x or derivatives thereof is penalized. For λ fixed, the solution to the optimization problem is straightforward to compute. For large scale problems, this is usually done via a Krylov subspace-based iterative projection method, such as CGLS, LSQR [25, 26] or LSMR [9]. At each iteration, these iterative methods project the original problem (1.2) onto Krylov subspaces of increasing dimension, they only require one matrix-vector product with each of the $A, A^{\mathrm{T}}, L, L^{\mathrm{T}}$, and they have short term recurrences (so storage is minimal). Provided that the number of iterations is small relative to the dimension of the problem, and that fast matrix-vector products are possible, these are relatively efficient solvers.

The so-called hybrid methods [3, 4, 8, 12, 13, 21, 23] exploit the efficiency of iterative solvers such as those mentioned in the previous paragraph to select a good regularization parameter when $R(x)$ has the form $\|Lx\|_2^2$. Iterative solvers produce a small-dimensional projected problem that retains certain features of the original large-scale problem. The order of the projected problem equals the number of performed iterations. Additional regularization (for instance, Tikhonov regularization) is then efficiently applied to the small-dimensional projected problem. From there, a regularized solution to the original large-scale problem can be obtained. The upside of the hybrid approach is that the regularization parameter is selected at each iteration for the small-dimensional problem only: in this setting, some well-known techniques such as the discrepancy principle or the $\mathcal{L}$-curve criterion can be applied with a negligible computational cost (both in terms of floating point operations and storage), by manipulating the much smaller projected quantities. However, with the usual choices of L mentioned above, the solutions will be smoother than desired.

Two well-known alternative choices for $R(x)$ when one desires regularized solutions with sharp edges are $\mathrm{TV}(x)$ (isotropic total variation) and $\|Lx\|_p^p$, with L the discrete gradient operator and p close to 1 (see [32] and references therein). For a single fixed λ, the algorithms employed to solve the optimization problem (1.2) usually require much more computational effort than if the two-norm regularization is used. Moreover, the biggest pitfall of using these regularization operators is that λ is not known *a priori*. Thus, many optimization problems need to be solved for a discrete set of λ in order to compute a meaningful solution.

A natural and well-established way of handling problem (1.2) with $R(x) = \|Lx\|_p^p$, $p \neq 2$, is to reformulate it as a sequence of quadratic problems: the ℓth quadratic problem involves a regularization term of the form $R(x) = \|M^{(\ell)}x\|_2^2$, where the ℓth regularization matrix $M^{(\ell)}$ is defined with respect to L and the approximate solution of the $(\ell - 1)$th quadratic problem in the sequence. Here and in the following we refer to this approach as iteratively reweighted norm (IRN) method: this was first proposed in [15], and then extended in [28, 29, 34]. Recently, an Arnoldi–Tikhonov (hybrid) method for $R(x) := \|x\|_1$ was proposed [12], which can be used when the operator A is square. This approach combines a reweighting strategy (to approximate $R(x)$) and a hybrid strategy applied with the discrepancy principle to choose the regularization parameter. Since the weights are updated at each iteration of the hybrid method, and since the weights are formally regarded as variable preconditioners after transforming the current quadratic Tikhonov problem into standard form, the reweighted problem is efficiently projected using the flexible Arnoldi algorithm: because of this, only one cycle of iterations should be performed, while, for all the other IRN methods mentioned so far, inner–outer iteration cycles are needed when dealing with large-scale problems. This method based on flexible Krylov subspaces has been extended to work with $R(x) = \mathrm{TV}(x)$ in [11], and with rectangular matrices A as well as a sparsity-under-transform regularization term in [7].

In this paper we propose a technique that is similar to the IRN methods, in the sense that we, too, generate a sequence of quadratic problems, with a different weight matrix at each outer-iteration, and employ a hybrid approach on each regularized problem in the sequence. However, our method differs from each of the methods listed above in that: (a) our regularization matrix is different—indeed it is rectangular and possibly not full column rank; (b) the operator A

may be rectangular; (c) due to (a) and (b), we must use a different hybrid projection algorithm. More precisely, we propose an automated, inner–outer iterative approach, consisting in solving a sequence of regularized least squares problems with two-norm regularization of the form

$$\min_{x\in\Gamma^{(\ell)}} \|Ax-b\|_2^2+\lambda^2\|M^{(\ell)}x\|_2^2, \quad \ell=1,2,3,\ldots \tag{1.3}$$

where a near-optimal value of λ for the ℓth problem (1.3), which we denote as $\lambda_{*,\ell}$, is determined on-the-fly for each regularization operator $M^{(\ell)}$. Here, $\Gamma^{(\ell)}$ denotes a specific projection space in which we will look for a projected solution. Because this solution space depends in part on $M^{(\ell)}$ itself (but not on the regularization parameter) we use a superscript on it as well. To compute $\Gamma^{(\ell)}$ and $\lambda_{*,\ell}$ we employ the hybrid regularization approach (inner iterations) of [20]. We give a method for designing $M^{(\ell)}$ adaptively (outer iterations) in such a way that edges are enhanced as ℓ increases. We discuss expected behavior on a class of images, and present results and comparisons on applications in x-ray CT and image deblurring to demonstrate that our algorithm has the potential to produce high-quality images in a computationally efficient manner.

This paper is organized as follows. In section 2, we give definitions, notations, and motivations for studying our new inner–outer iterative approach. We describe hybrid iterative regularization algorithms in section 3, highlighting the particular method in [20] that we leverage in our new approach. In section 4, we develop our inner–outer iterative algorithm, and provide some analysis. Section 5 is devoted to numerical results. We give conclusions and outline future work in section 6.

2. Background and motivation

We begin by motivating the need to take $M^{(\ell)}$ in (1.3) as something other than the identity I or the discrete gradient operator. Then, we discuss other algorithms proposed in the literature that have been used for edge-preservation.

2.1. Filtering methods

Consider the model (1.1) and assume that A has rank ρ. Consider the singular value decomposition (SVD) of $A=U\Sigma V^{\mathrm{T}}=\sum_{i=1}^{\rho}\sigma_i u_i v_i^{\mathrm{T}}$. Then, the minimum norm least squares solution to $Ax=b$ is easily seen to be given by

$$x=\sum_{i=1}^{\rho}\frac{u_i^{\mathrm{T}}b}{\sigma_i}v_i=\sum_{i=1}^{\rho}\left(\frac{u_i^{\mathrm{T}}b_{\mathrm{true}}}{\sigma_i}v_i+\frac{u_i^{\mathrm{T}}\eta}{\sigma_i}v_i\right), \tag{2.1}$$

and $\sigma_1\gg\sigma_\rho>0$. Under the assumption of white noise η, the values of $|u_i^{\mathrm{T}}\eta|$ are approximately constant, whereas the $|u_i^{\mathrm{T}}b_{\mathrm{true}}|$ are large and dominant for small indices i but, eventually, decay toward zero and the noise becomes dominant (this property in commonly known as discrete Picard condition, see [17]).

Regularization methods like TSVD, and Tikhonov with $R(x)=\|x\|_2^2$ in (1.2), work by damping the contribution from the terms corresponding to large indices in (2.1), i.e., they are essentially filtering methods and compute a regularized solution of the form

$$x_{\mathrm{reg}}=\sum\phi_i\frac{u_i^{\mathrm{T}}b}{\sigma_i}v_i, \tag{2.2}$$

where ϕ_i denotes a scalar that decreases with increasing i; see [17, 19]. However, the edge content from the image is encoded in the high-frequency terms that are damped or discarded in (2.2), and hence such regularized solutions tend to be smooth.

Next, let us consider what happens when $R(x) = \|Lx\|_2^2$, where L is the discrete gradient operator. To fix notation, we assume the following. Let X be an $N_\mathrm{v} \times N_\mathrm{h}$ image, and let $x = \mathrm{vec}(X)$ be its vectorized version, obtained by stacking its columns. Define

$$L_\mathrm{v} = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & \cdots & 0 & -1 \end{bmatrix}$$

to be the $N_\mathrm{v} - 1 \times N_\mathrm{v}$ discrete first derivative operator. We define $L_\mathrm{h} \in \mathbb{R}^{(N_\mathrm{h}-1)\times N_\mathrm{h}}$ similarly. Using $\otimes$ to represent Kronecker product between two matrices, note that

- $L_v X = \mathrm{reshape}((I \otimes L_\mathrm{v})\mathrm{vec}(X)) \in \mathbb{R}^{(N_\mathrm{v}-1)\times N_\mathrm{h}}$ is the image that approximates all vertical derivatives;
- $XL_\mathrm{h}^\mathrm{T} = \mathrm{reshape}((L_\mathrm{h} \otimes I)\mathrm{vec}(X)) \in \mathbb{R}^{N_\mathrm{v}\times(N_\mathrm{h}-1)}$ is the image that approximates horizontal derivatives.

Here $\mathrm{reshape}(\cdot)$ can be regarded as the inverse of the $\mathrm{vec}(\cdot)$ operator. Now let

$$L = \begin{bmatrix} I \otimes L_\mathrm{v} \\ L_\mathrm{h} \otimes I \end{bmatrix}. \tag{2.3}$$

The matrix L is called the discrete gradient operator and, clearly, $Lx \in \mathbb{R}^{N_\mathrm{h}(N_\mathrm{v}-1)+(N_\mathrm{h}-1)N_\mathrm{v}}$ contains the edge information of x.

Consider (1.2) with $R(x) = \|Lx\|_2^2$. Since Lx contains the edge information, adding a multiple of $R(x)$ to the data misfit component of the cost functional has the effect of producing regularized solutions where jumps are penalized. For images that are supposed to be smooth, this is not a problem. But assume for a moment that X is a piecewise constant image with jump discontinuities. Additive noise has the effect of creating jumps, so $R(x)$ smooths out the discontinuities in a region of otherwise constant value, which is desired. Unfortunately, because the values of Lx are large near true jump discontinuities, the price one pays is that the jump discontinuity is also smoothed out.

2.2. Edge preservation algorithms

As mentioned in section 1, two popular choices for $R(x)$ in (1.2) are total variation [30] and $\|Lx\|_p^p$ for $1 \leqslant p < 2$ [17]. The total variation functional can be expressed, in a discrete setting, as

$$R(x) = \mathrm{TV}(x) = \sum_{i=1}^{N_\mathrm{h}-1} \sum_{j=1}^{N_\mathrm{v}-1} \sqrt{|L_\mathrm{v}X|_{ij}^2 + |XL_\mathrm{h}^\mathrm{T}|_{ij}^2}. \tag{2.4}$$

Clearly, either choice for R necessitates using an iterative optimization technique and, at each step of this routine, a linear system or least squares problem may be solved to generate the next search direction. The lagged diffusivity fixed point iteration (see [33] for algorithm details) is a popular choice, but other options such as steepest descent, Newton's method, and primal-dual approaches are also possible (see [32] for details and references). For $R(x) = \|Lx\|_1$, one can employ iteratively reweighted methods (from the Newton family) or interior point methods [22]. Other approximations to TV may also be employed (see, for example, [1] and the discussion in [32]).

However, a key issue that is somewhat disregarded within the edge-preserving solvers listed so far is the choice of regularization parameter λ. Indeed, the optimization codes written to solve (1.2) with the edge-preserving $R(x)$ outlined in the previous paragraph all assume that the value of λ is fixed. Since the optimal value of λ is not known a *priori*, the procedure usually advocated is to solve multiple optimization problems for different values of λ picked from a discrete set, and then choose the value of λ that gives a 'best' image according to some metrics. This involves many calls to the optimization routine; moreover, the convergence behavior of a particular run of the optimization method will vary with λ. Thus, on one side it would appear that recovery of edge information via any of the aforementioned methods would be very computationally expensive, because we have to solve multiple optimization problems for multiple λ (one for each value of λ); moreover, as hinted above, each one of the optimization problems may require multiple linear system solves to determine the outcome.

On the other side, if we take $R(x) = \|x\|_2^2$ or $R(x) = \|Lx\|_2^2$, the cost functional in (1.2) is quadratic and it is relatively straightforward to compute the solution to the optimization problem (i.e., in this case the cost of one full solve is roughly the cost of a single optimization step for the edge-preserving R's). Even so, especially when it comes to the choice of the regularization parameter, further savings in the case of a quadratic problem (1.2) can be obtained through hybrid iterative methods. Hybrid regularization methods operate by projecting a quadratic problem onto smaller-dimensional subspaces where certain properties of noise and signal separation can be preserved. When employing a hybrid approach, (additional) regularization is applied to the small projected problem instead, and the regularization parameter is chosen by applying a suitable heuristic to the projected problem. This is significantly cheaper than solving a quadratic problem (1.2) for each new value of λ. This is discussed in detail in the next section. However, adopting this family of quadratic $R(x)$'s, we trade edge content for computational efficiency.

In this paper we propose to merge the best of these two worlds, defining an adaptive sequence of quadratic problems

$$\min_x \|Ax - b\|_2^2 + \lambda^2 \|\underbrace{D^{(\ell)} L}_{=:M^{(\ell)}} x\|_2^2, \quad \ell = 1, 2, 3, \ldots$$

whereby a hybrid approach can be employed to automatically choose the regularization parameter in an efficient way for each quadratic problem in the sequence. Our choice of $M^{(\ell)}$ has features in common with the methods proposed in [2, 34], in that it involves different weightings of the gradient operator to emphasize edges in current estimated solutions. However, it differs from those methods in that the effects of our $M^{(\ell)}$ are cumulative, i.e., information from all the previous steps is retained. To run our algorithm we only need to be able to perform matrix vector products with the forward operator A in (1.1) and with L in (2.3), and their respective transposes. Moreover, our approach uses hybrid methods to automatically determine regularization parameters.

3. Hybrid iterative methods

In this section, we give the general outline for two (hybrid) approaches that could ultimately be used within our inner–outer iterative strategy. The first one is based on LSQR, while the second one is based on a joint bidiagonalization algorithm. In our problems, $D^{(\ell)}L$ will not have full column rank, which makes the second option a more elegant choice and easier to implement (for this reason, the second option is used to perform the numerical experiments displayed in section 5). However, for completeness and ease of exposition, we present the LSQR-based hybrid approach first. For more specific details, and for additional methods from the hybrid regularization family, we refer to [12, 13].

Consider the problem

$$\min_x \|Ax - b\|_2^2 + \lambda^2 \|Mx\|_2^2, \tag{3.1}$$

where, for the remainder of this section, we assume that M remains fixed. This problem is mathematically equivalent to the linear least squares problem

$$\min_x \left\| \begin{bmatrix} A \\ \lambda M \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2. \tag{3.2}$$

The hybrid methods described in the following subsections compute approximate solutions to (3.1) and (3.2) as functions of λ, and select good choices of λ in efficient ways.

3.1. Hybrid-LSQR method for standard form Tikhonov

Let us first address solving (3.1) iteratively by LSQR [25, 26]. Implicitly, this approach amounts to transforming problem (3.1) into standard form and constraining the solution to belong to a k-dimensional subspace $\mathcal{K}_k$ at the kth iteration, i.e., solving

$$\min_{y \in \mathcal{K}_k} \|AM_A^\dagger y - b\|_2^2 + \lambda^2 \|y\|_2^2, \tag{3.3}$$

where $M_A^\dagger$ is the A-weighted pseudo-inverse of M, defined as

$$M_A^\dagger = \left(I - \left(A\left(I - M^\dagger M\right)\right)^\dagger A\right) M^\dagger. \tag{3.4}$$

In the above definition, the use of the dagger symbol without an accompanying subscript (e.g., $M^\dagger$) is meant to denote the standard Moore–Penrose pseudo-inverse (see, for instance, [14]). Note that, if M is invertible, then $M_A^\dagger = M^{-1}$, and, if M has full column rank, then $M_A^\dagger = M^\dagger$. If M is an underdetermined matrix, then $M_A^\dagger$ may not be the same as $M^\dagger$. For further details on A-weighted pseudo-inverses in the context of Tikhonov regularization, we refer to [17].

When adopting LSQR, $\mathcal{K}_k$ represents the k-dimensional Krylov subspace

$$\mathcal{K}_k = \text{span}\{(AM_A^\dagger)^{\mathrm{T}} b, ((AM_A^\dagger)^{\mathrm{T}}(AM_A^\dagger))(AM_A^\dagger)^{\mathrm{T}} b, \ldots, ((AM_A^\dagger)^{\mathrm{T}}(AM_A^\dagger))^{k-1}(AM_A^\dagger)^{\mathrm{T}} b\},$$

generated by $(AM_A^\dagger)^{\mathrm{T}}(AM_A^\dagger)$ and $(AM_A^\dagger)^{\mathrm{T}} b$. The kth iteration of the LSQR algorithm updates the matrix recurrence

$$(AM_A^\dagger)V_k = U_{k+1}B_k$$

where B_k is $(k+1) \times k$ lower bidiagonal, V_k has k orthonormal columns that span $\mathcal{K}_k$, and U_{k+1} has $k+1$ orthonormal columns with $U_{k+1}(\beta e_1) = b$. Taking $y = V_k z$ (i.e., imposing $y \in \mathcal{K}_k$), and exploiting unitary invariance of the two-norm allows us to rewrite (3.3) as the $O(k)$-sized least squares problem

$$\min_{z \in \mathbb{R}^k} \|B_k z - \beta e_1\|_2^2 + \lambda^2 \|z\|_2^2, \tag{3.5}$$

which is a Tikhonov-regularized projected problem in standard form. Denoting by $z^{(\lambda,k)}$ the minimizer of the above problem, $y^{(\lambda,k)} = V_k z^{(\lambda,k)}$ is the minimizer of (3.3). If k is large enough, so that the spectral behavior of the bidiagonal matrix B_k mimics that of the operator $AM_A^{\dagger}$, then the impact of the choice of λ on the quality of the solution can be observed. However, as hinted in section 1, k should be assumed small enough, so that problem (3.5) can be efficiently generated and solved. For fixed k, we can use any suitable parameter choice strategy (such as the discrepancy principle or the $\mathcal{L}$-curve criterion) on the projected problem (3.5) to choose λ.

Still assuming k fixed, and regardless of the parameter selection method, let λ_k denote the estimated optimal parameter. Assuming k is large enough, we then find $z^{(\lambda_k,k)}$ by solving (3.5), we set $y^{(\lambda_k,k)} = V_k z^{(\lambda_k,k)}$, and we transform from standard form (3.3) back to general form (3.1), to get an approximation $x^{(\lambda_k,k)}$ of an optimal regularized solution of (3.1). Computing $y^{(\lambda_k,k)}$ may be done with short term recurrences on the iteration k (i.e., in k, for fixed λ), without storage of V_k. See, for example, [21] for details.

To evaluate the computational cost of this class of hybrid algorithms we should consider that each iteration requires a matrix-vector product with $AM_A^{\dagger}$ and one with its transpose. Note that $AM_A^{\dagger}v$ is computed as the two subsequent 'products' $A(M_A^{\dagger}v)$. The product $M_A^{\dagger}v$ is actually computed implicitly: for instance, if $M_A^{\dagger} = M^{-1}$, then $M^{-1}v = g$ is the same as $v = Mg$, so in practice the system $Mg = v$ is solved for g, and M^{-1} is never explicitly computed. Thus, one usually refers to the 'product' of $M_A^{\dagger}v$ as 'applying $M_A^{\dagger}$'. Unfortunately, applying $M_A^{\dagger}$ in the case of the A-weighted pseudo-inverse defined as in (3.4) requires additional solver calls. Moreover, in general, each solver call can involve invoking iterative solvers. Therefore, analyzing the computational cost of the products $M_A^{\dagger}v$, and of this hybrid algorithm as a whole, may be difficult. In any case, the overall cost in producing a regularized solution depends predominantly on k, and the cost of an application of $M_A^{\dagger}$ and matrix-vector products with A (likewise, their transposes); we must highlight again that the computational cost is essentially independent of the number of discrete values of λ tested to determine the optimal regularization parameter.

3.2. Hybrid method for general form Tikhonov

Applying $M_A^{\dagger}$ is often non-trivial [17], especially when M is defined as a product of matrices (see also [11]). The authors of [20] develop an alternative hybrid method based on a joint bidiagonalization algorithm, which we now describe. Here, M can be rectangular and does not need to be full rank. Avoiding transformation into standard form (and the need of applying $M_A^{\dagger}$), and starting instead from (3.2), the formulation

$$\min_{x\in\mathcal{Z}_k}\left\|\begin{bmatrix}A\\ \lambda M\end{bmatrix}x-\begin{bmatrix}b\\0\end{bmatrix}\right\|_2,\quad \mathcal{Z}_k=\operatorname{span}\{z_1,\ldots,z_k\},\tag{3.6}$$

is considered, where a partial joint bidiagonalization for A and M is updated at each iteration as follows

$$AZ_k = U_{k+1}B_k,\qquad MZ_k = \hat{U}_k\hat{B}_k,\tag{3.7}$$

where $Z_k = [z_1,\ldots,z_k]$, $U_{k+1}(\beta e_1) = b$, $\beta = \|b\|_2$. Taking $x = Z_k w$ in (3.6) results in the following equivalent problems

$$\min_{x\in\mathcal{Z}_k}\left\|\begin{bmatrix}A\\ \lambda M\end{bmatrix}x-\begin{bmatrix}b\\0\end{bmatrix}\right\|_2=\min_{w\in\mathbb{R}^k}\left\|\begin{bmatrix}AZ_k\\ \lambda MZ_k\end{bmatrix}w-\begin{bmatrix}b\\0\end{bmatrix}\right\|_2=\min_{w\in\mathbb{R}^k}\left\|\begin{bmatrix}B_k\\ \lambda\hat{B}_k\end{bmatrix}w-\begin{bmatrix}\beta e_1\\0\end{bmatrix}\right\|_2.$$

The last problem in the above equations is a Tikhonov-regularized problem of dimension $O(k)$ involving only bidiagonal matrices and, for k small relative to the original problem dimension, it is significantly cheaper to solve. Let us denote by $w^{(\lambda,k)}$ the minimizer of this $O(k)$-dimensional problem: then $x^{(\lambda,k)} = Z_k w^{(\lambda,k)}$ is the minimizer of (3.6). The bidiagonal structure leads to recurrence updates (in k, for fixed λ) for $x^{(\lambda,k)}$, for the residual norm $\|Ax^{(\lambda,k)} - b\|_2$ and for the regularization norm term $\|Mx^{(\lambda,k)}\|_2$, as functions of λ. So, for each fixed k, the regularization parameter can be computed efficiently using strategies such as the discrepancy principle or the $\mathcal{L}$-curve criterion, which only require those quantities.

Thus, as long as k is large enough that the bidiagonal pair inherits certain spectral properties of the original operators, the impact of λ as a regularization parameter for the kth system can be observed, a suitable parameter λ_k can be selected, and the solution $x^{(\lambda_k,k)} = Z_k w^{(\lambda_k,k)}$ of (3.6) can be readily obtained by employing short term recurrences: this solution is an estimate of an optimal regularized solution of (3.2). The reader is referred to [20] for further details.

The algorithm requires only that products with A, M and their transposes can be computed. The cost of generating the bidiagonal pair is somewhat difficult to quantify. The kth step requires one call to LSQR to produce orthogonal projections that are needed to update the partial joint bidiagonalization (3.7), and each LSQR iteration requires 4 matrix-vector products (with each of $A, A^{\mathrm{T}}, M, M^{\mathrm{T}}$). If LSQR needs m_k steps, then the cost of a call to LSQR is proportional to m_k times the cost of those 4 matrix-vector products. The value of m_k is typically small relative to problem dimension but, depending on k, it may affect the overall reconstruction time. Promising methods for keeping m_k small are currently under investigation. Nevertheless, bounds on the behavior of k and m_k are possible, and for some classes of operators (e.g., see the CT image examples in the section 5) both k and m_k are very small. Even in cases where k and/or m_k are larger, when this approach is used as part of the inner–outer edge-preserving iterative scheme below, the overall performance in producing quality images without any parameter tuning favors our approach versus multiple calls per λ to sophisticated optimization routines, whose behavior and performance is at least as difficult to analyze.

4. New inner–outer iterative algorithm for edge preservation

So far, we have seen that hybrid iterative methods for two-norm Tikhonov-regularized problems are computationally appealing for two reasons: (a) their cost is associated with the number of iterations, k, and the ability to compute matrix-vector products, thus requiring little storage (although a precise computational cost estimate can be difficult; see sections 3.1 and 3.2); (b) a good value for λ can be computed on-the-fly essentially for free, with strategies like the discrepancy principle or the $\mathcal{L}$-curve criterion. On the other hand, choosing the two-norm of the gradient as a regularizer has a smoothing effect; changing to the p-norm of the gradient, or changing to TV regularization, would enhance edges but requires the solution of a sequence of difficult optimization problems for many values of λ.

We therefore look to build an approach that allows us to leverage the capabilities of hybrid algorithms, with their computational efficiency and ability to choose regularization parameters on-the-fly, but has the adaptivity to capture edge information in the reconstruction process. Clearly, based on the discussion in section 2, this means we cannot use a regularization term of the form $R(x) = \|Lx\|_2^2$.

We propose to solve the sequence of problems

$$x^{(*,k_\ell)} = \arg\min_{x\in\Gamma_{k_\ell}^{(\ell)}} \|Ax - b\|_2^2 + \lambda_{*,\ell}^2 \|D^{(\ell)}Lx\|_2^2, \quad \ell = 1, 2, 3, \ldots \tag{4.1}$$

where $\lambda_{*,\ell}$ denotes the 'optimal' regularization parameter for the ℓth problem, chosen according to some criterion. The ℓth problem computes a solution over a subspace of dimension k_ℓ (i.e., subspace dimension can vary with ℓ, hence the subscript on Γ). In this section, we will answer the following questions:

- How can $D^{(\ell)}$ be expressed, and how should it evolve to produce edge-enhanced images?
- How can $\lambda_{*,\ell}$ be estimated efficiently?
- How can $x^{(*,k_\ell)}$ be obtained efficiently?

4.1. Diagonal weighting

Suppose $x^{(*,0)}$ is an estimate of x. Assume further that it is a good enough estimate that at least one edge is visible. If we compute the gradient image (as a matrix-vector product) and normalize it, i.e.,

$$g^{(0)} = |Lx^{(*,0)}|/\|Lx^{(*,0)}\|_\infty, \tag{4.2}$$

where $|\cdot|$ in the numerator is used to denote element-wise absolute value, then this vector will have the largest values equal to 1 where the dominant edges are located (these are precisely the values we *do not want to penalize*) and smaller nonnegative values where pixels are still corrupted by noise (these are precisely the values we *do want to penalize*). If we now consider the 'image'

$$d^{(1)} := \mathbf{1} - (g^{(0)}) \,\hat{\bullet}\, q, \quad q > 0, \tag{4.3}$$

where $\mathbf{1}$ is the vector of all ones and $\hat{\bullet}$ denotes element-wise exponentiation, then the values we do not want to penalize have been mapped to the smallest values $\geqslant 0$, and the noisy parts of the image we want to penalize have been mapped to the largest values $\leqslant 1$. Choosing $q \gg 1$ in (4.3) results in more penalization of the supposedly smooth regions (as more entries in the vector $d^{(1)}$ are forced to be close to 1), and therefore in an overall smoother reconstruction. Choosing $q \ll 1$ in (4.3) results in less penalization of the supposedly smooth regions (as less entries in the vector $d^{(1)}$ are forced to be close to 1), and therefore in an overall less smooth reconstruction. Thus if, for $\ell = 1$, we use

$$D^{(1)} := \mathrm{diag}(d^{(1)})$$

in (4.1), this has the effect of enforcing smoothness on parts of the image where we still expect to be able to see, and improve, smoothness; at the same time, it does not wash out the 'true' edges visible in $x^{(*,0)}$, because we have (almost) zeroed out those components of the gradient image, as prescribed by the diagonal weighting in (4.1).

Suppose for the moment that $\lambda_{*,1}$ is known, and that $x^{(*,1)}$ has been determined (as described in the next subsections). If $x^{(*,1)}$ is an improvement over $x^{(*,0)}$, specifically containing new edge information, then the reweighted and normalized gradient image $g^{(1)} = |D^{(1)}Lx^{(*,1)}|/\|D^{(1)}Lx^{(*,1)}\|_\infty$ should reveal the new edge information (as $g^{(1)}$ will have the largest values close to 1 where the newly reconstructed edges are located, and smaller nonnegative values where the edges are already present in $x^{(*,0)}$ and the smooth parts of $x^{(*,1)}$ are located). We set $d^{(2)} = \mathbf{1} - (g^{(1)}) \,\hat{\bullet}\, q$, so that again the new edge information we do not want to penalize has been mapped to the smallest values $\geqslant 0$, and the noise-contaminated components and the former edges have been mapped to the largest values $\leqslant 1$. We then take

$$D^{(2)} = \mathrm{diag}(d^{(2)})D^{(1)}.$$

The entries of $D^{(2)}$ still remain between 0 and 1. Notice that $D^{(2)}$ encodes information about both the previous solution estimates $x^{(*,0)}$ and $x^{(*,1)}$ and, in particular, their edges that will not be penalized; indeed, the null space of $D^{(2)}L$ is spanned by the constant image **1**, and by the piecewise-constant images having edges corresponding to the dominant edges of $x^{(*,0)}$ and $x^{(*,1)}$. This process is iterated so that, at the ℓth outer iteration, one takes

$$g^{(\ell-1)} = \frac{|D^{(\ell-1)}Lx^{(*,\ell-1)}|}{\|D^{(\ell-1)}Lx^{(*,\ell-1)}\|_\infty}, \quad d^{(\ell)} := 1 - g^{(\ell-1)} \,\hat{\bullet}\, q, \quad D^{(\ell)} = \mathrm{diag}(d^{(\ell)})D^{(\ell-1)}. \quad (4.4)$$

More insight about the specific choice (4.4) is given in [31]; this is essentially related to an update rule stemming from an alternating minimization scheme applied to an objective function involving the sum of a fit-to-data least-squares term, a reweighted regularization term that enhances edges, and the generalized Kullback–Leibler divergence that enforces proximity of two consecutive weights. More generally, our new method can be regarded within the framework of iterative refinement algorithms that aim at improving the performance of their non-iterative counterparts; see, e.g., [24] for the total variation case. However, the definition of the weights $D^{(\ell)}$ (leading to the solution of a sequence of linear problems depending on all the previous approximate reconstructions, and depending on the functional expression of $d^{(\ell)}$) are specific of our new method. Note also that, in the framework of PDE methods for imaging, the general idea of enhancing edges by introducing some adaptive anisotropy (e.g., some sort of reweighting dependent on the gradient of the image) can be linked back to the classical Perona–Malik nonlinear diffusion model (see, e.g., [5, chapter 4]). More algorithmic details are unfolded in the following sections 4.2 and 4.3.

4.2. Choosing the regularization parameter and computing the solution

We now further elaborate on techniques for the computation of the regularization parameters and the corresponding solutions for the ℓth problem in (4.1).

We can use a hybrid method to determine $\lambda_{*,\ell}$ and $x^{(*,k_\ell)}$. In our numerical examples, we use the method described in section 3.2, together with either the discrepancy principle or the $\mathcal{L}$-curve criterion for the projected problem (3.6). The following statements can be easily reformulated to work in connection with the hybrid method described in section 3.1.

The discrepancy principle can be applied if a good estimate of the norm $\|\eta\|_2$ of the noise affecting the data is available. When used at the kth iteration of the hybrid solver for (4.1), it consists in determining the parameter λ_k such that

$$\|B_k w^{(\lambda_k,k)} - \beta e_1\|_2 = \tau\|\eta\|_2 = \|Ax^{(\lambda_k,k)} - b\|_2, \quad (4.5)$$

where $\tau > 1$, $\tau \simeq 1$, is a safety threshold, and where the factorizations (3.7) and the properties of the matrices appearing therein have been exploited. Equation (4.5) is nonlinear with respect to λ_k, and an appropriate zero-finder should be employed (see, for instance, [27]). Problem (4.5) can be solved at a negligible additional computational cost, provided that k is relatively much smaller than the original problem size $\min\{n, m\}$, and by updating some relevant factorizations for B_k as k increases. The inner iterations (in k) are stopped when no significative variations in λ_k are detected for two consecutive values of k. Alternatively, one can employ the so-called 'secant method' [13], which updates λ_k in such a way that stopping by the discrepancy principle is ensured. When the stopping criterion is satisfied, we take $k_\ell = k$, and the optimal parameter $\lambda_{*,\ell}$ for the ℓth problem in (4.1) is set to λ_{k_ℓ}. Also, $\Gamma^{(\ell)}_{k_\ell} = \mathcal{Z}_{k_\ell}$, and $x^{(*,k_\ell)} = Z_{k_\ell} w^{(k_\ell,\lambda_{k_\ell})}$.

The $\mathcal{L}$-curve criterion can be employed at the kth iteration of the hybrid solver for (4.1), even when an estimate for $\|\eta\|_2$ is not available: in this case, the horizontal axis is determined by $\|B_k w^{(\lambda,k)} - \beta e_1\|_2 = \|Ax^{(\lambda,k)} - b\|_2$, and the vertical axis is determined by

Algorithm 1. Inner–outer iterations for recovering $N_{\mathrm{v}} \times N_{\mathrm{h}}$ images.

```
Input an initial guess $x^{(*,0)}$ for the solution. Set $D^{(0)} = I$. Choose $q > 0$ for (4.3).
For $\ell = 1, \dots,$ until a stopping criterion is satisfied
  Set $g^{(\ell-1)} = |D^{(\ell-1)}Lx^{(*,\ell-1)}|/\|D^{(\ell-1)}Lx^{(*,\ell-1)}\|_\infty$
  Set $d^{(\ell)} := 1 - g^{(\ell-1)} \widehat{\bullet} q$ (produces a map to [0, 1])
  Set $D^{(\ell)} = \mathrm{diag}(d^{(\ell)})D^{(\ell-1)}$
  Set $M^{(\ell)} = D^{(\ell)}L$
  Run the hybrid algorithm for problem (4.1) to choose $\lambda_{*,\ell}$ and return $x^{(*,k_\ell)}$.
end
Return $x_{\mathrm{reg}} = x^{(*,\ell)}$.
```

$\|\hat{B}_k w^{(\lambda,k)}\|_2 = \|M^{(\ell)}x^{(\lambda,k)}\|_2$, where $M^{(\ell)} = D^{(\ell)}L$; the above quantities are evaluated, at each iteration k, for a given fixed discrete set of values of λ, which correspond to points on the $\mathcal{L}$-curve. The inner iterations (in k) are stopped when the corner of an $\mathcal{L}$ is visible for a few iterations, and the estimated corner value is constant for a few iterations. When this stopping criterion is satisfied, we take $k_\ell = k$, $\Gamma_{k_\ell}^{(\ell)} = \mathcal{Z}_{k_\ell}$, and $\lambda_{*,\ell} = \lambda_{k_\ell}$ as the optimal parameter for the ℓth problem (4.1). The solution $x^{(*,\ell)}$ is generated by a short term recurrence, and it is therefore very storage efficient. We refer to [20] for further computational details.

We stress that somewhat different computational approaches should be adopted when considering the discrepancy principle or the $\mathcal{L}$-curve criterion within the inner hybrid solver for the ℓth problem in (4.1). This is because, at the kth hybrid iteration, the former computes approximate solutions that correspond to regularization parameters that are sequentially selected by the zero finder applied to (4.5), while the latter simultaneously computes approximate solutions that correspond to a predetermined set of regularization parameters. However, in both cases, the computational overhead of selecting $\lambda_{*,\ell}$ is negligible if $k_\ell \ll \min\{m, n\}$.

Similarly to the discrepancy principle and the $\mathcal{L}$-curve criterion, every parameter choice rule that relies on the current residual or solution (semi)norms is an attractive option for hybrid methods.

Clearly, this process of adaptively computing optimal solutions and updating the regularization operator to account for newly acquired edge information can be repeated for the $(\ell + 1)$th problem in (4.1). A sketch of the resulting algorithm is presented in the next section.

4.3. Summary of the main algorithm

As already highlighted, our new algorithm iteratively improves available solution estimates by recovering and exploiting edge information on the go. This algorithm is inherently based on an inner–outer iteration scheme, where inner iterations are handled with a hybrid algorithm, and the outer iterations update the available regularization term. The main operations performed by the new method are summarized in algorithm 1.

The outer iterations in algorithm 1 need a starting guess $x^{(*,0)}$ to compute the first weighting matrix $D^{(1)}$. Although algorithm 1 can accept any $x^{(*,0)}$ (even possibly determined by a different regularization method), a natural choice is to take $x^{(*,0)}$ as a constant vector: in this way, for $\ell = 1$, we get $Lx^{(*,\ell-1)} = 0$ so that $D^{(1)} = I$ is the identity matrix of order $(N_{\mathrm{h}}(N_{\mathrm{v}} - 1) + N_{\mathrm{v}}(N_{\mathrm{h}} - 1))$, and the first problem in the sequence of quadratic problems (4.1) is actually a Tikhonov regularization problem with $R(x) = \|Lx\|_2^2$. A constant $x^{(*,0)} = 0$ will be used for the numerical experiments in section 5.

Two appropriate stopping criteria should be set when implementing algorithm 1. The first one prescribes how to terminate the hybrid iterations, and should be devised in connection with the regularization parameter choice strategy, as discussed in section 4.2 (basically, we monitor the stabilization of the approximate λ_k along consecutive hybrid iterations). The second one prescribes how to terminate the outer iterations: since we would ideally like to iterate until there is no more real edge information to recover, we track $\|Lx^{(*,\ell)}\|_2$. If we detect that this quantity is decreasing at step ℓ, then we have likely oversmoothed our solution, so we break out of the (outer) loop.

4.4. Analyzing the main algorithm

Directly from the definition of the diagonal weights given in section 4.1, it is immediate to state that:

(a) If a diagonal entry of $D^{(\ell)}$ becomes 0, it stays 0 at future outer iterations.

(b) The ith diagonal entry of two consecutive weight matrices is such that

$$[D^{(\ell)}]_{ii} \leqslant [D^{(\ell-1)}]_{ii}. \tag{4.6}$$

(c) Large values in the gradient images incur smaller weights, which are even smaller when $p \gg 1$ is selected.

To further analyze the behavior of algorithm 1, we ignore the subspace constraint $\Gamma^{(\ell)}_{k_\ell}$ in (4.1) and consider the sequence of minimization problems

$$x^{(\lambda,\ell)} = \arg\min_x \|Ax - b\|_2^2 + \lambda_\ell^2 \|D^{(\ell)}Lx\|_2^2 = \min_x\ J_{\lambda,\ell}(x), \quad \ell = 1, 2, 3, \ldots. \tag{4.7}$$

This assumption simplifies the following derivations, and it is also quite realistic because the hybrid solutions of (4.1) resemble the minimizers of (4.7) for k sufficiently large (i.e., when enough inner hybrid iterations are performed).

Now let us give a preliminary qualitative overview of the role played by the regularization parameter λ_ℓ in (4.7). For smaller λ_ℓ, the model fidelity through the operator $A^{\mathrm{T}}A$ is the dominant term in the cost functional. For larger λ_ℓ the regularization (derivative) term is dominant. Note that solving problem (4.7) is also mathematically equivalent to solving

$$(A^{\mathrm{T}}A + \lambda_\ell^2 L^{\mathrm{T}}(D^{(\ell)})^2 L)x = A^{\mathrm{T}}b. \tag{4.8}$$

In this reformulation it is easier to see that the second term plays the role of a diffusion-like operator (see, for instance, [2, 33]). Diffusion operators are frequently used in denoising (see, for instance, [6] and references therein), and so we may also state that a large λ_ℓ favors denoising. When $D^{(1)} = I$ (e.g., when algorithm 1 runs with a constant initial guess $x^{(*,0)}$, the applied regularization just enforces smoothness on the derivative. In our problems, we know that the image is not smooth, so overly smooth solution estimates result in a large data-fidelity mismatch. Therefore, any reasonable parameter selection routine, would not select a λ_1 too large, since to do so would return a smooth solution at the price of a large residual; this is surely not the case when the discrepancy principle is adopted. However, if $x^{(*,0)}$ is already quite successful in revealing the edges of x_{true}, then, for any vector v of appropriate size,

$$\|D^{(1)}Lv\|_2 < \|Lv\|_2,$$

so that, when running the second outer iteration of algorithm 1 (i.e., $\ell = 2$), a minimizer of (4.7) with a larger λ_2 does not enforce smoothness uniformly across the entire solution, and could be tolerated. Indeed, the second outer iteration behaves more like denoising than deblurring, because the weights are large in regions where the edges were not detected, and those are regions where a good amount of smoothing is still needed. Thus, we expect the parameter selection strategy to choose a larger value of λ_2. We expect this to be a common trend also in the following outer iterations, meaning that there is increased reliance on the denoising provided by the regularization term as the outer iterations progress. We can prove that this is exactly the case when the discrepancy principle is used to select λ_ℓ for the ℓth problem in (4.7). This result (stated in proposition 4.1) requires a fair amount of derivations, and it is preceded by a lemma (lemma 4.1) that states that the discrepancy curves, i.e., the curves

$$(\lambda, \|b - Ax^{(\lambda,\ell)}\|_2), \quad \lambda \geqslant 0 \tag{4.9}$$

decrease with respect to the outer iteration counter ℓ.

Lemma 4.1. *Assume that the null spaces of A and $D^{(\ell)}L$ intersect trivially, for $\ell = 1, 2, \ldots$. Let $r^{(\lambda,\ell)} = b - Ax^{(\lambda,\ell)}$ be the discrepancy associated to the ℓth problem in the sequence (4.7), with $\lambda_\ell = \lambda \geqslant 0$, $\ell = 1, 2, \ldots$. Then*

$$\|r^{(\lambda,\ell)}\|_2 \geqslant \|r^{(\lambda,\ell+1)}\|_2. \tag{4.10}$$

Proof. Consider the ℓth problem in the sequence (4.7). We first have to introduce some notations. Let

$$\widehat{A}^{(\ell)} = A^{\mathrm{T}}A + \lambda^2 L^{\mathrm{T}}(D^{(\ell)})^2 L,$$

i.e., the matrix appearing on the left-hand side of (4.8), with $\lambda_\ell = \lambda \geqslant 0$. Then

$$\|r^{(\lambda,\ell)}\|_2^2 = b^{\mathrm{T}}(I - A(\widehat{A}^{(\ell)})^{-1}A^{\mathrm{T}})^2 b.$$

Assume that $[D^{(\ell)}]_{i_\ell,i_\ell} \neq [D^{(\ell+1)}]_{i_\ell,i_\ell}$, $i_\ell \in \{1_\ell, \ldots, h_\ell\} \subset \{1, \ldots, 2(n - \sqrt{n})\}$, $1_\ell < 2_\ell < \cdots < h_\ell$, i.e., the diagonal weighting matrices computed at two consecutive iterations differ at most for $h \geqslant 0$ entries. Then, thanks to (4.6), we can write

$$(D^{(\ell)})^2 = (D^{(\ell+1)})^2 + \sum_{i=1}^{h} (d_{i_\ell})^2 e_{i_\ell} e_{i_\ell}^{\mathrm{T}}, \quad d_{i_\ell} > 0.$$

Let

$$(D^{(\ell),j})^2 = (D^{(\ell+1)})^2 + \sum_{i=1}^{j} (d_{i_\ell})^2 e_{i_\ell} e_{i_\ell}^{\mathrm{T}} = (D^{(\ell),j-1})^2 + (d_{j_\ell})^2 e_{j_\ell} e_{j_\ell}^{\mathrm{T}}, \quad j = 0, \ldots, h$$

(so that $(D^{(\ell),0})^2 = (D^{(\ell+1)})^2$, and $(D^{(\ell),h})^2 = (D^{(\ell)})^2$). Correspondingly, let

$$r^{(\lambda,\ell),j} = b - Ax^{(\lambda,\ell),j}, \quad \text{where} \quad x^{(\lambda,\ell),j} = (\widehat{A}^{(\ell),j})^{-1}A^{\mathrm{T}}b,\ \widehat{A}^{(\ell),j} = A^{\mathrm{T}}A + \lambda^2 L^{\mathrm{T}}(D^{(\ell),j})^2 L.$$

Note that

$$\widehat{A}^{(\ell),j} = \widehat{A}^{(\ell),j-1} + \lambda^2 (d_{j_\ell})^2 L^{\mathrm{T}} e_{j_\ell} (L^{\mathrm{T}} e_{j_\ell})^{\mathrm{T}}, \quad j = 1, \ldots, h, \tag{4.11}$$

i.e., the difference between two consecutive matrices of the form $\widehat{A}^{(\ell),j}$ is a symmetric rank-1 matrix. Let us fix $j \in \{0, \ldots, h-1\}$. If we show that

$$\|r^{(\lambda,\ell),j+1}\|_2^2 \geqslant \|r^{(\lambda,\ell),j}\|_2^2, \tag{4.12}$$

we can conclude that $\|r^{(\lambda,\ell)}\|_2^2 := \|r^{(\lambda,\ell),h}\|_2^2 \geqslant \|r^{(\lambda,\ell),h-1}\|_2^2 \geqslant \ldots \geqslant \|r^{(\lambda,\ell),0}\|_2^2 =: \|r^{(\lambda,\ell+1)}\|_2^2$. Thanks to the Sherman–Morrison formula (see, e.g., [14]),

$$(\widehat{A}^{(\ell),j+1})^{-1} = (\widehat{A}^{(\ell),j})^{-1} - (\widehat{A}^{(\ell),j})^{-1} w w^{\mathrm{T}} (\widehat{A}^{(\ell),j})^{-1}, \quad \text{where}$$

$$w = \frac{\lambda d_{j_\ell} L^{\mathrm{T}} e_{j_\ell}}{\sqrt{1 + \lambda^2 d_{j_\ell}^2 (L^{\mathrm{T}} e_{j_\ell})^{\mathrm{T}} (\widehat{A}^{(\ell),j})^{-1} (L^{\mathrm{T}} e_{j_\ell})}}$$

and where we have used twice that $(\widehat{A}^{(\ell),j})^{-1}$ is symmetric positive definite. Then

$$\begin{aligned}
\|r^{(\lambda,\ell),j+1}\|_2^2 &= b^{\mathrm{T}} \Big(I - A((\widehat{A}^{(\ell),j})^{-1} - (\widehat{A}^{(\ell),j})^{-1} w w^{\mathrm{T}} (\widehat{A}^{(\ell),j})^{-1}) A^{\mathrm{T}} \Big)^2 b \\
&= b^{\mathrm{T}} (I - A(\widehat{A}^{(\ell),j})^{-1} A^{\mathrm{T}} + \widehat{w}\widehat{w}^{\mathrm{T}})^2 b \\
&= \underbrace{b^{\mathrm{T}} (I - A(\widehat{A}^{(\ell),j})^{-1} A^{\mathrm{T}})^2 b}_{=\|r^{(\lambda,\ell),j}\|_2^2} + \underbrace{2 b^{\mathrm{T}} \widehat{w}\widehat{w}^{\mathrm{T}} (I - A(\widehat{A}^{(\ell),j})^{-1} A^{\mathrm{T}}) b}_{=:\delta^{(1)}} \\
&\quad + \underbrace{b^{\mathrm{T}} \widehat{w}((\widehat{w})^{\mathrm{T}}(\widehat{w}))\widehat{w}^{\mathrm{T}} b}_{=:\delta^{(2)}}
\end{aligned}$$

where we have defined $\widehat{w} = A(\widehat{A}^{(\ell),j})^{-1} w$. Here $\delta^{(2)} \geqslant 0$, since $\widehat{w}\widehat{w}^{\mathrm{T}}$ is symmetric positive semi-definite. If we show that $\delta^{(1)} \geqslant 0$, then we have (4.12) and therefore (4.10). Let

$$A = U^{(\ell),j} \Sigma^{(\ell),j} (X^{(\ell),j})^{-1}, \qquad D^{(\ell),j} L = V^{(\ell),j} \Gamma^{(\ell),j} (X^{(\ell),j})^{-1}$$

be the GSVD of the matrix pair $(A, D^{(\ell),j} L)$, [14]. Then (exploiting the properties of the matrices involved in the decompositions above), (4.11), and the definitions of w and $\widehat{w}$,

$$\begin{aligned}
\delta^{(1)} &= \underbrace{\frac{2\lambda^2 (d_{j_\ell})^2}{1 + \lambda^2 d_{j_\ell}^2 (L^{\mathrm{T}} e_{j_\ell})^{\mathrm{T}} (\widehat{A}^{(\ell),j})^{-1} (L^{\mathrm{T}} e_{j_\ell})}}_{=:\mu>0} b^{\mathrm{T}} A (\widehat{A}^{(\ell),j})^{-1} L^{\mathrm{T}} e_{j_\ell} e_{j_\ell}^{\mathrm{T}} \\
&\quad \times L (\widehat{A}^{(\ell),j})^{-1} A^{\mathrm{T}} (I - A(\widehat{A}^{(\ell),j})^{-1} A^{\mathrm{T}}) b \\
&= \mu \overbrace{b^{\mathrm{T}} U^{(\ell),j}}^{=:\widehat{b}^{\mathrm{T}}} \underbrace{\Sigma^{(\ell),j} \big((\Sigma^{(\ell),j})^2 + \lambda^2 (\Gamma^{(\ell),j})^2 \big)^{-1}}_{=:D_1^{(\ell),j}} \overbrace{X^{\mathrm{T}} L^{\mathrm{T}} e_{j_\ell}}^{=:\widehat{d}} e_{j_\ell}^{\mathrm{T}} L X D_2^{(\ell),j} \\
&\quad \times (U^{(\ell),j})^{\mathrm{T}} b = \mu \widehat{b}^{\mathrm{T}} D_1^{(\ell),j} \widehat{d}\widehat{d}^{\mathrm{T}} D_2^{(\ell),j} \widehat{b},
\end{aligned}$$

where

$$D_2^{(\ell),j} = ((\Sigma^{(\ell),j})^2 + \lambda^2 (\Gamma^{(\ell),j})^2)^{-1} \Sigma^{(\ell),j} \left(I - (\Sigma^{(\ell),j})^2 ((\Sigma^{(\ell),j})^2 + \lambda^2 (\Gamma^{(\ell),j})^2)^{-1} \right).$$

Note that both $D_1^{(\ell),j}$ and $D_2^{(\ell),j}$ are diagonal matrices with positive diagonal entries. One can easily show that $D_1^{(\ell),j} \widehat{d}\widehat{d}^{\mathrm{T}} D_2^{(\ell),j}$ has at most rank-1, its only nonzero eigenvalue is $(D_2^{(\ell),j} \widehat{d})^{\mathrm{T}} (D_1^{(\ell),j} \widehat{d}) > 0$ (since we are summing positive quantities), and is diagonalizable. Therefore it is positive semi-definite and $\delta^{(1)} \geqslant 0$, which concludes our proof. □

Proposition 4.1. *Assume that the null spaces of A and $D^{(j)} L$ intersect trivially, for $j = 1, 2, \ldots$, and assume that the regularization parameter λ_ℓ in (4.7) is determined according to the discrepancy principle. Then*

$$\lambda_\ell \geqslant \lambda_{\ell-1}, \quad \ell = 1, 2, 3, \ldots. \tag{4.13}$$

Proof. Take $\ell \geqslant 1$. Thanks to (4.10), the discrepancy curve (4.9) computed for the ℓth problem lays below the discrepancy curve computed for the $(\ell - 1)$th problem. Also, one can easily prove that, for a fixed ℓ and a variable $\lambda \geqslant 0$, the discrepancy curve increases with respect to λ. Since applying the discrepancy principle consists in solving

$$\|b - Ax^{(\lambda,\ell)}\|_2 = \tau\|\eta\|_2,$$

this implies (4.13). □

More generally, we experimentally find that the parameters $\lambda_{*,\ell}$ (often) satisfy $\lambda_{*,\ell} \geqslant \lambda_{*,\ell-1}$ also when inner hybrid iterations are provided (i.e., when problem (4.1) is solved instead of (4.7)), and that this also happens when the $\mathcal{L}$-curve criterion is used instead of the discrepancy principle (see section 5).

We conclude this section by stressing again that algorithm 1 works well when the initial solution estimates $x^{(*,0)}$ or $x^{(*,1)}$ are able to recognize at least one decent edge: if the initial estimates are too smooth, which might be the case for example in deblurring with a large noise to signal ratio, our method can fail, as the mapping from the normalized gradient image to [0, 1] (described in equation (4.3)) may be totally inappropriate. The qualitative performance of algorithm 1 also depends on the success of the parameter choice strategy but, as we show in the next section, it works quite well for many applications, providing significant enhancement in a completely automated fashion over the initial solution estimate.

5. Numerical results

All experiments were done in Matlab version 9.1. The synthetic test problems come from the software packages IR tools [10] and AIR tools II [18]; also a tomography experiment with publicly available real data from [16] is considered[6].

In this section, we would like to assess the potentialities of our new algorithm, and we mainly do so by comparing it to other similar inner–outer iteration methods for total variation regularization. Namely, we consider the IRN method for (isotropic) total variation (IRN–TV) proposed in [34]: similarly to our new method, IRN–TV updates some weights, defines a sequence of least squares problems (outer iterations), and solves each iteratively reweighted least squares problem iteratively (inner iterations). IRN–TV and our new method differ in the way the weights are defined. The IRN–TV weights $D^{(\ell)}$ for the ℓth least-squares problem (4.1) have the following expression

$$\begin{aligned} D^{(\ell)} &= W(x^{(*,\ell-1)}) = I_2 \otimes (W_R^{(\ell)})^{1/2}, \\ W_R^{(\ell)} &= \mathrm{diag}\left(\mathrm{vec}(L_{\mathrm{v}}X^{(*,\ell-1)})^2 + \mathrm{vec}(X^{(*,\ell-1)}L_{\mathrm{h}}^{\mathrm{T}})^2\right)^{-1/2}, \end{aligned} \tag{5.1}$$

where $x^{(*,\ell-1)}$ is the solution of the $(\ell-1)$th least squares problem and I_2 is the identity matrix of size 2. Fixing $y = \mathrm{vec}(Y)$, $Y \in \mathbb{R}^{N_{\mathrm{h}}\times N_{\mathrm{v}}}$, and evaluating the above weights in y leads to $\|W(y)Ly\|_2^2 = \mathrm{TV}(y)$, where L is the discrete gradient operator (2.3), and $\mathrm{TV}(\cdot)$ is defined as in (2.4). Because of this, we can expect that the reweighted regularization term in IRN–TV is a good approximation of $\mathrm{TV}(x_{\mathrm{true}})$ when $x^{(*,\ell-1)}$ is a good approximation of x_{true}. Moreover, when $[\mathrm{vec}(L_{\mathrm{v}}X)^2 + \mathrm{vec}(XL_{\mathrm{h}}^{\mathrm{T}})^2]_i$ is big (typically when there is an edge between the ith and the $(i+1)$th pixel), then $[(W_R^{(\ell)})^{1/2}]_{ii}$ is small (so that the corresponding gradient component is not

[6] MATLAB implementations of our methods, which should be used jointly with IR tools, are available at https://github.com/silviagazzola.

much penalized in the minimization process); vice versa, when $[\mathrm{vec}(L_{\mathrm{v}}X)^2 + \mathrm{vec}(XL_{\mathrm{h}}^{\mathrm{T}})^2]_i \simeq 0$ (typically corresponding to smooth portions of X) then $[(W_R^{(\ell)})^{1/2}]_{ii}$ is huge (so that the corresponding gradient component is penalized even more in the minimization process). Note that some thresholding of the weights may be necessary to avoid division by zero. IRN–TV and our new method also differ in the way each least squares problem is solved: IRN–TV assumes that a good value of the regularization parameter is fixed, and employs CGLS for the inner iterations. When it comes to the update of the weights, it should be noted that our new weights (4.4) penalize vertical and horizontal derivatives differently, while the same is not true for the IRN–TV weights (5.1). For this reason, for some test problems we also perform comparisons with an IRN scheme whose weights approximate a so-called anisotropic total variation regularization term, i.e., $R(x) = \|Lx\|_1$ in (1.2). Even if anisotropic TV is not considered in [34], the IRN–TV method can be easily modified to work in this setting by taking the following weights for the ℓth least-squares problem (4.1),

$$D^{(\ell)} = W(x^{(*,\ell-1)}) = \mathrm{diag}\left(|Lx^{(*,\ell-1)}|\right)^{-1/2}, \tag{5.2}$$

where L is the discrete gradient operator (2.3). We will use the acronym IRN–aTV to refer to the IRN method equipped with the weights (5.2). Note that, as in the IRN–TV case, some thresholding should be considered to safeguard against division by zero in (5.2).

We also make comparisons with two inner–outer iterative schemes that can be collocated somewhere in between IRN–TV or IRN–aTV, and our new algorithm. Namely: (a) we define the weights at the beginning of each outer iteration as in (5.1) or (5.2), and we solve each quadratic problem by the hybrid method described in section 3.2, with the added benefit of adaptively choosing the regularization parameter; (b) we define the weights at the beginning of each outer iteration as in section 4.1, and we solve each quadratic problem by CGLS, having fixed a regularization parameter. We also emphasize that, in order to improve the conditioning of the innermost iterations for algorithm 1, in all the experiments the rescaled matrix $10L$ is considered instead of the matrix L appearing in (2.3); see also [20] for more details.

5.1. Examples from x-ray CT

Parallel-beam CT. We use IR tools to setup the following x-ray tomography simulation:

- Generate a simulated true phantom image called ‘grains’ from AIR tools II, of size 128 × 128 pixels (shown in the leftmost frame of figure 3).
- Construct noise-free projections, along with the matrix A that simulates the ray trace forward operator, assuming a parallel beam x-ray transmission, with data collected at angles $0, 2, \ldots, 130$ degrees.
- Add 0.1% normally distributed (white Gaussian) noise.

More specifically, the test problem is generated with the following MATLAB statements:

```
ProblemOptions = PRset('angles', 0:2:130, 'phantomImage', 'grains');
[A, b_true, x_true, ProblemInfo] = PRtomo(128, ProblemOptions);
b = PRnoise(b_true, 1e-3);
```

First of all, we test our new algorithm for different parameter choice strategies employed within the inner hybrid scheme for generalized Tikhonov regularization (section 3.2). More precisely, we consider the discrepancy principle and the $\mathcal{L}$-curve criterion. The left frame of figure 1 shows a plot of different discrepancy curves defined, analogously to (4.9), as the graphs of the function $\|b - Ax^{(\lambda,k_\ell)}\|_2$ versus λ, and corresponding to different outer iterations ℓ. The

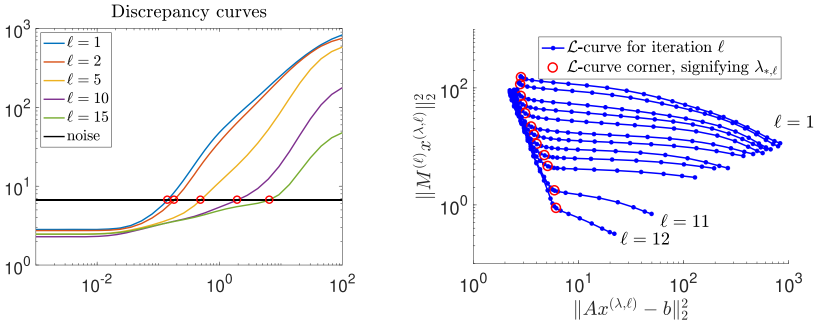

Figure 1. Parallel-beam CT test problem. Left frame: logarithmic plot of the discrepancy curves, i.e., $\|Ax^{(\lambda,k_\ell)} - b\|_2$ versus λ, at the end of some selected cycles of inner iterations (as implied from the text in the legend, these correspond to the outer iterations $\ell = 1, 2, 5, 10, 15$); the red circles denote the intersection between each discrepancy curve and the noise level line, which corresponds to the chosen regularization parameter, $\lambda_{*,\ell}$, for the particular outer iteration. Right frame: logarithmic plot of the $\mathcal{L}$-curves for each outer iteration (as implied from the text in the plot, the top curve corresponds to the first outer iteration, $\ell = 1$, and the curves below this correspond sequentially to iterations $\ell = 2, 3, \ldots, 11, 12$); the red circles denote corners of each $\mathcal{L}$-curve, which corresponds to the chosen regularization parameter, $\lambda_{*,\ell}$, for the particular outer iteration.

right frame of figure 1 shows a plot of the different $\mathcal{L}$-curves obtained at the end of each inner iteration cycle. Looking at these graphs, we can see that both the discrepancy curves and the $\mathcal{L}$-curves are nested, in agreement with the analysis performed in section 4.4. The behavior shown in figure 1 is shared by the other test problems considered in this section, and therefore similar graphs will not be fully reported again in the following.

When our algorithm is implemented with the stopping criterion described in section 4.3 (i.e., stopping as soon as $\|Lx^{(*,\ell)}\|_2$ decreases during two consecutive outer iterations) we have termination after $\ell = 15$ iterations if the discrepancy principle is adopted in the inner iterations, and after $\ell = 12$ iterations if the $\mathcal{L}$-curve criterion is adopted in the outer iterations.

Figure 2 shows a plot of the relative errors and chosen regularization parameters at each outer iteration, when the discrepancy principle and the $\mathcal{L}$-curve criterion are used to select the regularization parameter during the inner hybrid iterations. Note that, as expected, the regularization parameters increase as the outer iterations proceed: this illustrates the property derived in section 4.4 for the discrepancy principle, which experimentally holds also for the $\mathcal{L}$-curve criterion. This behavior of the regularization parameter is meaningful and desirable: indeed, as the outer iterations increase, we are recovering approximate solutions of enhanced quality that result in improved weights for the regularization term, which in turn should be weighted more to achieve reconstructions of even higher quality.

Computed reconstructions for the first outer iteration (that is, $x^{(*,1)}$), and for the final outer iteration (that is, $x^{(*,15)}$ or $x^{(*,12)}$, depending on the parameter choice strategy) are shown in figure 3. As we can see from these plots, as the outer iterations proceed, there is a significant improvement in the reconstructions, and in particular the edges at the final outer iteration are much sharper than in the initial outer iteration. Also, pixel intensities are very close to those of the true image.

Since our new algorithm is an inner–outer iterative strategy, and since so far only the behavior of the solution and regularization parameter across the outer iterations has been

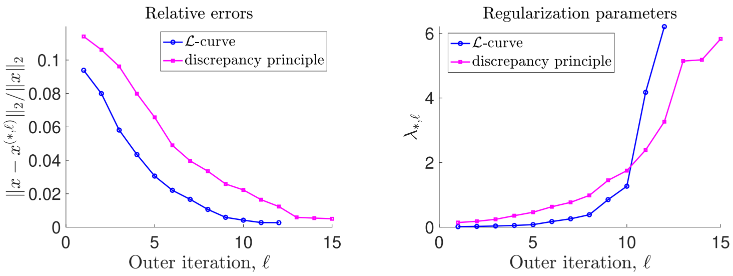

Figure 2. Parallel-beam CT test problem. Relative errors and regularization parameters values at each outer iterations ℓ, until the stopping criterion is satisfied. Both the discrepancy principle and the $\mathcal{L}$-curve criterion are considered.

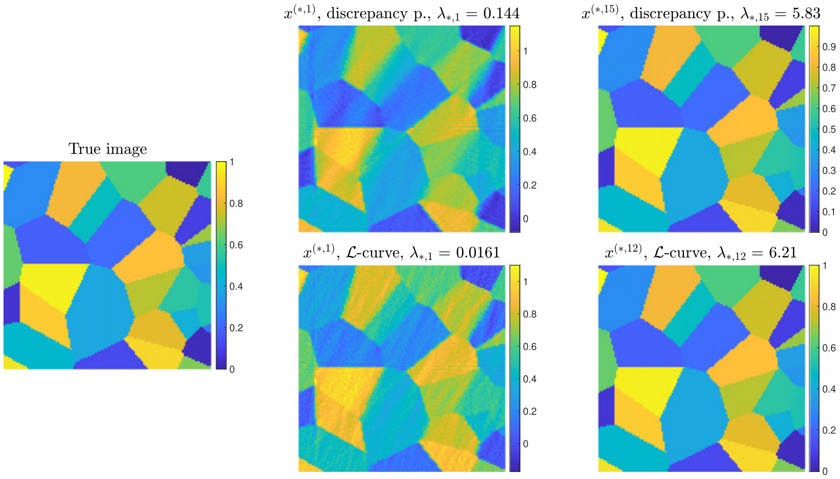
True image
1
0.8
0.6
0.4
0.2
0
$x^{(*,1)}$, discrepancy p., $\lambda_{*,1} = 0.144$
1
0.8
0.6
0.4
0.2
0
$x^{(*,15)}$, discrepancy p., $\lambda_{*,15} = 5.83$
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
$x^{(*,1)}$, $\mathcal{L}$-curve, $\lambda_{*,1} = 0.0161$
1
0.8
0.6
0.4
0.2
0
$x^{(*,12)}$, $\mathcal{L}$-curve, $\lambda_{*,12} = 6.21$
1
0.8
0.6
0.4
0.2
0

Figure 3. Parallel-beam CT test problem. Left frame: true phantom image x. Middle frames: reconstructions at the initial outer iteration. Right frames: reconstructions at the final outer iterations. The corresponding regularization parameters chosen by the hybrid method are displayed above each reconstruction. Solutions computed by the discrepancy principle and the $\mathcal{L}$-curve are displayed above and below, respectively.

displayed, figure 4 displays the relative errors against the number of total iterations. The end of each inner iteration cycle is marked by an asterisk. We can see that each inner iteration cycle consists of 60 iterations, i.e., the maximum allowed number of inner iterations: despite the inner stopping criterion for the inner iterations not being very effective for this example, we can observe that the quality of the reconstruction stabilizes and is almost optimal after some inner iterations (although it seems to slightly deteriorate for the discrepancy principle): this means that both adaptive parameter choice strategies are quite effective. Although not reported in the following, the other test problems considered in this section display a similar behavior.

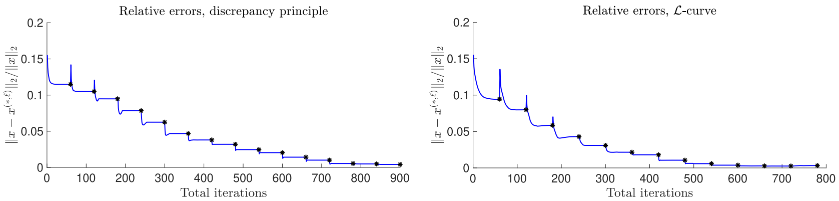

Figure 4. Parallel-beam CT test problem. Relative errors versus number of total iterations, when the discrepancy principle (left frame) and the $\mathcal{L}$-curve (right frame) are used to select the regularization parameter at each inner iteration. Black asterisks mark the last iteration of each inner cycle.

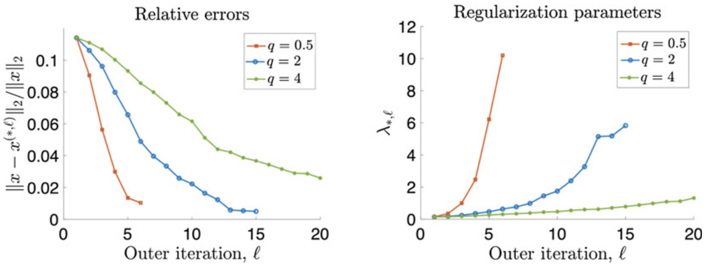

Figure 5. Parallel-beam CT test problem. Relative errors and regularization parameter versus number of outer iterations for three different choices of the power q in (4.3).

The definition of the weights at each outer iteration depends on a power $q > 0$ (see equation (4.4)). In section 4.1 we made the argument that the choices $q \ll 1$ and $q \gg 1$ correspond to less and more smooth reconstructions, respectively. We now experimentally assess how the value of q affects the quality of the reconstructions when the discrepancy principle is employed to adaptively set the regularization parameter at each inner iteration (specifically for the present test problem; similar features are shared by the other test problems). Figure 5 shows the values of the relative errors and the regularization parameter versus the number of outer iterations for three values of q; $q = 2$ is the value selected to display the previous graphs. Zoom-ins of the reconstructed images are displayed in figure 6, also as surfaces. Although all the considered values of q deliver reconstructions of excellent quality, we can see some slight spurious oscillations in the supposedly constant patches reconstructed taking $q = 0.5$, and we can clearly see that some of the edges reconstructed using $q = 4$ are washed out. Interestingly enough, still comparing the choices $q = 0.5$ and $q = 4$ in figure 5, we can see that, since $q = 0.5$ intrinsically enforces less smoothness through the weights, a larger regularization parameter is automatically set to compensate for this, and less outer iterations are performed; vice versa, since $q = 4$ intrinsically enforces more smoothness through the weights, a smaller regularization parameter is automatically set to compensate for it, and more outer iterations are performed. To generate the following graphs and when considering all the following test problems, we will always take $q = 2$.

We now consider some comparisons between the new solver and different inner–outer iterative methods for edge enhancement in imaging. First of all, we assess the effect of a different

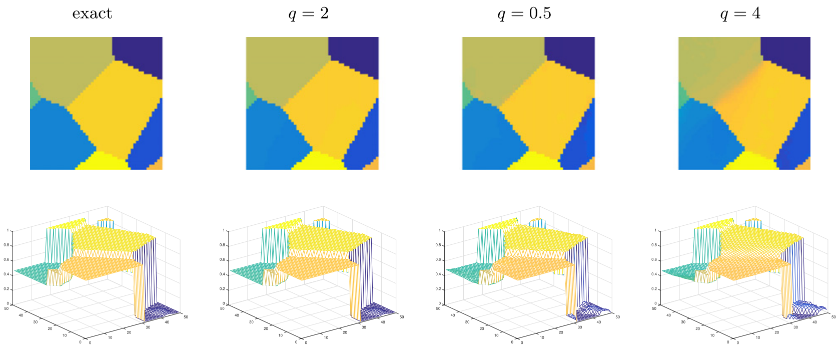

Figure 6. Parallel-beam CT test problem. Zoom-ins of the exact phantom and the reconstructions obtained when the outer iterations terminate, also displayed as surface plots.

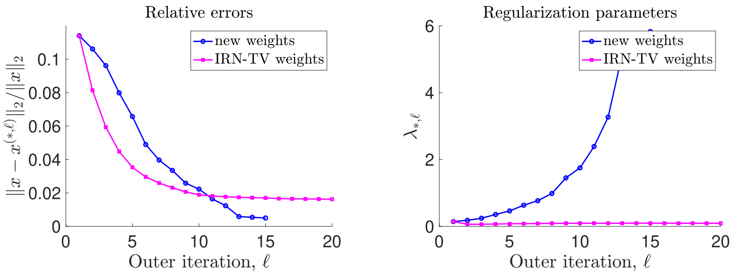

Figure 7. Parallel-beam CT test problem. Relative errors and regularization parameter versus number of (outer) iterations; both methods use the hybrid method for general form Tikhonov during the inner iterations, and adaptively select the regularization parameter according to the discrepancy principle.

choice of the weight matrix: namely, we keep an inner iteration scheme based on the hybrid solver for general form Tikhonov described in section 3.2, together with the discrepancy principle to adaptively set the regularization parameter, and we choose the weights as in (5.1), i.e., the weights associated to the IRN–TV methods. In figure 7 we display the behavior of the relative errors and regularization parameters versus the number of outer iterations, considering the new weights and the IRN–TV ones. We can clearly see that the IRN–TV method seems to perform well during the early (outer) iterations, but then the quality of the reconstructed solution rapidly stagnates, while the new method keeps improving. Also, the regularization parameter selected by the discrepancy principle seems to be almost constant right from the early outer iterations when the IRN–TV weights are used, while the regularization parameter selected by the same rule keeps increasing when the new weights are used. The substantially different orders of magnitude of the regularization parameters chosen for the two methods can be explained by noting that the weights used in algorithm 1 are rescaled, while the ones used by IRN–TV (5.1) are not. Correspondingly, figure 8 shows the reconstructions computed by the two methods at selected outer iterations: namely, at iterations $\ell = 2$ (i.e., as soon as the reweighting becomes

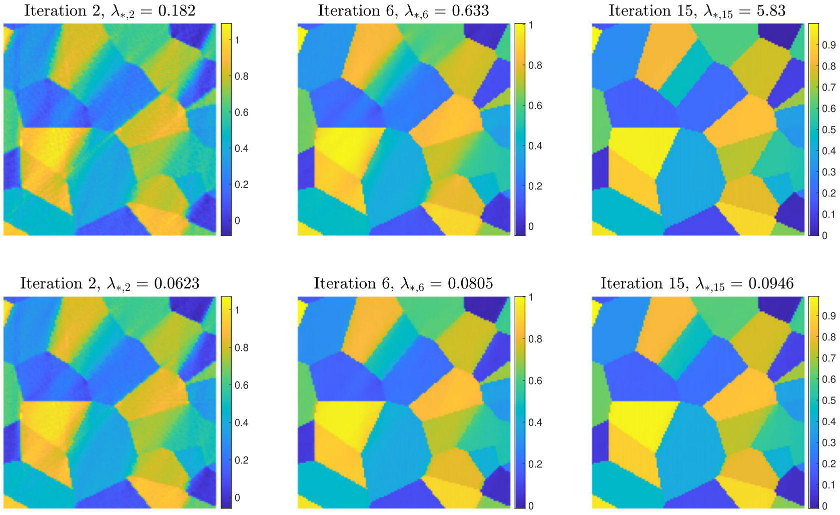
Iteration 2, $\lambda_{*,2} = 0.182$
Iteration 6, $\lambda_{*,6} = 0.633$
Iteration 15, $\lambda_{*,15} = 5.83$
Iteration 2, $\lambda_{*,2} = 0.0623$
Iteration 6, $\lambda_{*,6} = 0.0805$
Iteration 15, $\lambda_{*,15} = 0.0946$
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Figure 8. Parallel-beam CT test problem. Upper row: reconstructions obtained employing the new weights. Lower row: reconstructions obtained employing the IRN–TV weights. The reconstructions obtained at the outer iterations $\ell = 2, 6, 15$ are displayed, reporting also the value of the regularization parameter $\lambda_{*,\ell}$.

effective; see section 4.3), $\ell = 6$ (i.e., when the relative error associated to the IRN–TV weights is lower than the one obtained employing the new weights), and $\ell = 15$ (i.e., when the stopping criterion for the outer iterations is satisfied employing the new weights). We can clearly see that, although the reconstructions obtained using the IRN–TV weights improve to some extent as the outer iterations proceed, eventually the new method better recovers the edges and the piecewise constant features of the phantom. In order to better understand the reasons behind the different performances of these two inner–outer iterative schemes, in figure 9 we display the entries of the weighting diagonal matrices, in logarithmic scale and rearranged as images, at selected outer iterations. Note that, directly from the definition of the weights (see (4.4) and (5.1)), when considering the new weights it is meaningful to display two images (one for the weights applied to the vertical derivatives, and one for the weights applied to the horizontal derivatives), while only one image suffices when considering the IRN–TV weights (because the same weights containing information about both the vertical and the horizontal derivatives are applied to both the vertical and the horizontal derivatives). Also note that, still from the definition, the new weights are normalized while the IRN–TV ones are not. Looking at figure 9 it is evident that the new weights associated to both the vertical and horizontal derivatives correctly recover most of the edge locations (which are mapped to the smallest values $\geqslant 0$), and the smooth regions (which are mapped to the largest values $\leqslant 1$): in this way, appropriate penalization happens and the reconstructions as well as the weights improve along the outer iterations. The same is not true for the IRN–TV weights: while the locations of the edges are somewhat recovered (and the smallest $\geqslant 0$ weights are assigned to them), the smooth regions do not properly show up (and very oscillating weights are assigned to them): these weights are clearly not effective in enforcing piecewise smoothness, and almost no improvement can be eventually seen in both reconstructions and weights as the outer iterations proceed. Such behav-

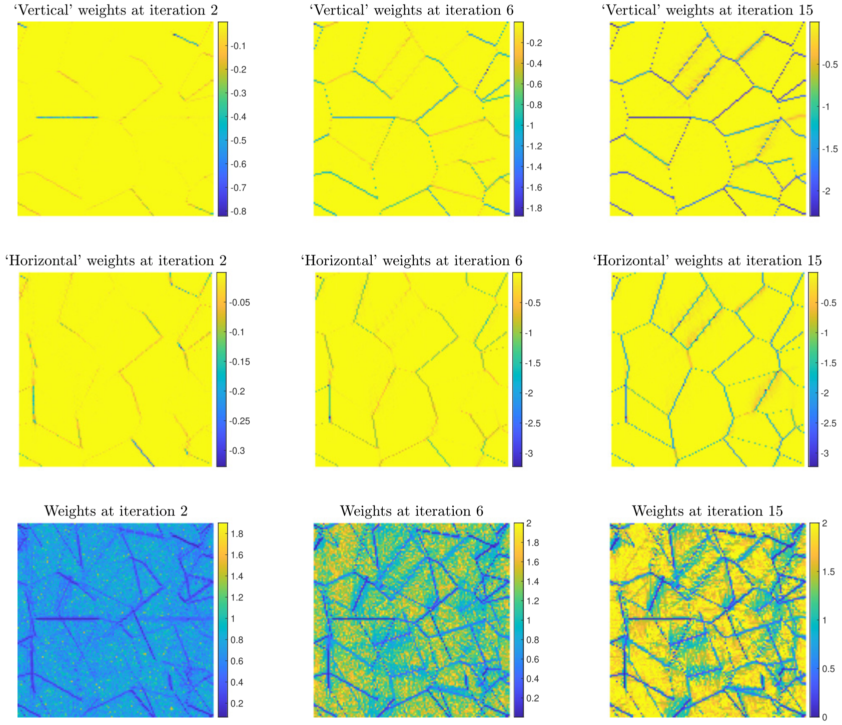

Figure 9. Parallel-beam CT test problem. Upper row: new weights, to be applied to the vertical derivatives. Middle row: new weights, to be applied to the horizontal derivatives. Lower row: IRN–TV weights. The weights used at the outer iterations $\ell = 2$ (leftmost column), 6 (middle column), 15 (rightmost column) are displayed in logarithmic scale.

ior of the weights can be observed in all the considered test problems, even if not reported in the following.

Finally, we show two more comparisons, that assess the influence of the inner iterative solver on the overall behavior of the method. Namely, we consider the inner–outer iterative schemes implemented with both the new and the IRN–TV weights, and with both the hybrid and the CGLS methods as inner solvers. Since CGLS requires a fixed value of the regularization parameter to be available in advance of each cycle of inner iterations, we first run the methods based on the hybrid solver for general form Tikhonov that adaptively chooses the regularization parameter at each inner iteration according to the discrepancy principle: in this way, a parameter $\lambda_{*,\ell}$ is eventually set at the ℓth outer iteration. When running the methods based on CGLS, we take $\lambda_{*,\ell}$ as regularization parameter for the ℓth outer iteration. Figure 10 displays the relative errors versus the number of outer iterations for the four instances of inner–outer iterative methods just described. In both the frames displayed in figure 10 we can see that the results obtained using CGLS as inner iterative solver are not generally as good as the ones obtained using the hybrid strategy: although there is not much difference when the new weights are used, the performance of CGLS with the IRN–TV weights is much worse than when the algorithm based on the hybrid method is used. This could be because $\lambda_{*,\ell}$

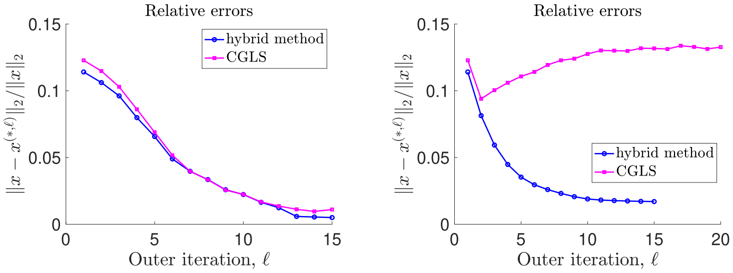

Figure 10. Parallel-beam CT test problem. Relative errors versus number of (outer) iterations for different inner solvers. Left frame: the new weights are used. Right frame: the IRN–TV weights are used.

is supposed to be a nearly-optimal regularization parameter when the joint bidiagonalization algorithm is used to project the ℓth quadratic problem (4.1); however CGLS projects the ℓth quadratic problem into a different subspace, and the same regularization parameter $\lambda_{*,\ell}$ might not be such a nearly-optimal choice for CGLS. Adaptively choosing the regularization parameter for the CGLS inner iterations might help improving the quality of the reconstructions. The behavior of CGLS shown in figure 10 can be consistently observed for all the test problems described below; therefore, to keep this section compact, such graphs will not be reported again.

Parallel-beam CT with very limited angle. We use IR tools to vary some of the options defining the previous x-ray tomography simulations. We still wish to consider a parallel beam x-ray transmission problem, but we would like to change the phantom image to the well-known Shepp–Logan one, of size 128 × 128 pixels (shown in the leftmost frame of figure 13). Moreover, we would like to make the reconstruction problem even more challenging by using heavily limited projection angles located at $0, 1, \ldots, 45$ degrees. Again, we generate this test problem within IR tools, using instructions similar to the previous ones, where a new `ProblemOptions` structure is defined in the following way:

```
ProblemOptions = PRset('angles', 0:45, 'phantomImage', 'shepplogan');
```

The measured data b is corrupted by Gaussian white noise of level 5×10^{-3}.

As in the previous example, we first run our algorithm with both the discrepancy principle and the $\mathcal{L}$-curve criterion to choose the regularization parameter within the inner hybrid scheme for generalized Tikhonov regularization. In both cases, the method performs the maximum allowed number of outer iterations $\ell = 20$. Figure 11 shows a plot of the relative errors and chosen regularization parameters at each outer iteration. We can clearly see that, when considering both the discrepancy principle and the $\mathcal{L}$-curve criterion, the reconstructions greatly improve as the outer iterations proceed, with the latter strategy being perhaps more efficient; however, it could be the case that the discrepancy principle achieves the same relative error as the $\mathcal{L}$-curve criterion if more outer iterations are performed. Note that, as expected, the regularization parameters selected by the discrepancy principle increase with the outer iterations; the same overall trend is recovered when the regularization parameters are selected by the $\mathcal{L}$-curve criterion, although there is a slightly more oscillating and stagnating behavior.

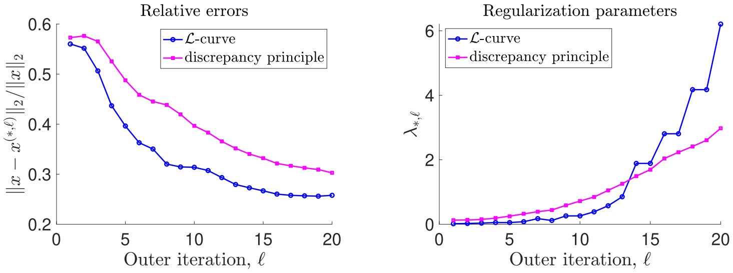

Figure 11. Parallel-beam CT *with very limited angle* test problem. Relative errors and regularization parameters values at each outer iteration ℓ. Both the discrepancy principle and the $\mathcal{L}$-curve criterion are considered.

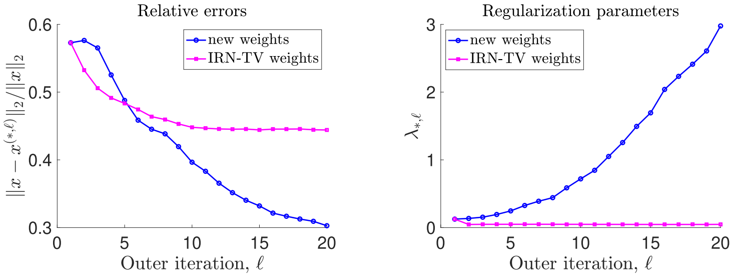

Figure 12. Parallel-beam CT with very limited angle test problem. Relative errors and regularization parameter versus number of (outer) iterations; both methods use the hybrid method for general form Tikhonov during the inner iterations, and adaptively select the regularization parameter according to the discrepancy principle.

Next, we compare the new method to the other inner–outer iterative method that still employs the hybrid solver for general form Tikhonov regularization to handle the inner iterations, while updating the IRN–TV weights (5.1) at each outer iteration. Figure 12 displays the relative error and regularization parameter values versus the number of outer iterations when the discrepancy principle is employed to adaptively choose the regularization parameter at each inner iteration of both methods. Similarly to the previous test problem, we can clearly see that, when the IRN–TV weights are used, the quality of the solution greatly improves in the first outer iterations but then stagnates. Instead, our new method delivers reconstructions of consistently increasing quality as the outer iterations progress, resulting in an overall more accurate reconstruction.

Finally, figure 13 displays the true phantom image along with some relevant reconstructions computed with the new and the IRN–TV weights. Namely, we show the reconstructions $x^{(*,2)}$ obtained as soon as the weights become active, and the reconstructions $x^{(*,20)}$, obtained when the iterations stop. We can clearly see that the edges at the final outer iteration are much sharper than at the initial outer iteration: this is especially true for the new weights, while some artefacts are evident in the IRN–TV reconstructions.

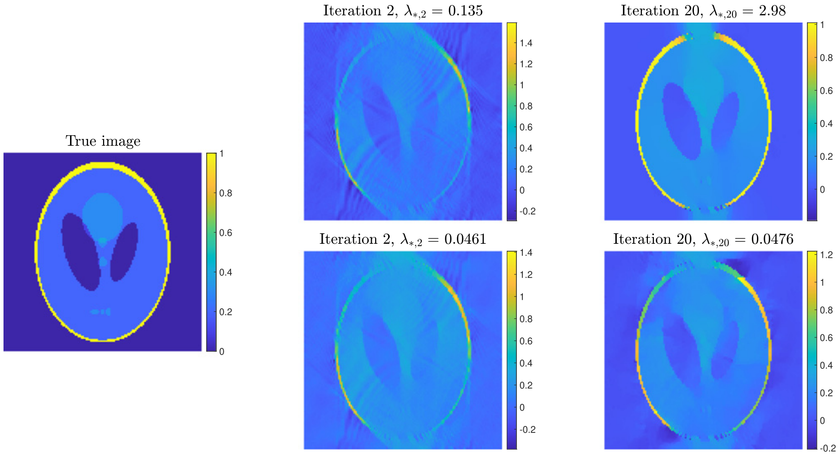
True image
1
0.8
0.6
0.4
0.2
0
Iteration 2, $\lambda_{*,2} = 0.135$
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Iteration 20, $\lambda_{*,20} = 2.98$
1
0.8
0.6
0.4
0.2
0
Iteration 2, $\lambda_{*,2} = 0.0461$
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Iteration 20, $\lambda_{*,20} = 0.0476$
1.2
1
0.8
0.6
0.4
0.2
0
-0.2

Figure 13. Parallel-beam CT with very limited angle test problem. Left frame: true phantom image x. Middle frames: reconstructions at the second outer iteration. Right frames: reconstructions at the final outer iteration. The corresponding regularization parameters chosen by the hybrid method with the discrepancy principle are displayed above each reconstruction. Solutions computed with the new and the IRN–TV weights are displayed above and below, respectively.

Undersampled fan-beam CT. This example considers a real tomographic scan of a walnut, using data that are publicly available; see [16]. More precisely, the goal is to reconstruct a central slice of a walnut at resolution 164 × 164 pixels, given 120 projections thereof. Since no information about the noise level in the data is (easily) available, we run our algorithm using the $\mathcal{L}$-curve criterion to set the regularization parameter within the inner hybrid scheme for generalized Tikhonov regularization. We perform 6 outer iterations, consisting of 100 inner iterations each. We provide comparisons with the inner–outer iterative strategy that uses the IRN–TV weights (5.1) and the hybrid solver for general form Tikhonov, equipped with the $\mathcal{L}$-curve criterion. Figure 14 displays some relevant reconstructions. We show the initial reconstructions $x^{(*,1)}$ obtained at the end of the first inner iteration cycle, when a Tikhonov-regularized problem with regularization term $R(x) = \|Lx\|_2^2$ is used (i.e., without reweighting), and the reconstructions $x^{(*,2)}$ and $x^{(*,6)}$ obtained when the new and the IRN–TV weights are included within the regularization term. We can clearly see that, when comparing $x^{(*,1)}$ and $x^{(*,2)}$ (computed with both the new and the IRN–TV weights), edges are more easily visible in the latter. Moreover, when the new weights are used, $x^{(*,6)}$ has a much more piecewise constant appearance than $x^{(*,2)}$ when the IRN–TV weights are used, $x^{(*,2)}$ and $x^{(*,6)}$ are not very different (although, upon a more careful inspection, the latter has less spurious oscillations in the pixel values). The left frame of figure 15 displays the values of the regularization parameter selected by the $\mathcal{L}$-curve criterion at the end of each inner iteration cycle, when both the new and the IRN–TV weights are used. For the new weights we can observe that the regularization parameter does not always increase as the outer iterations progress: indeed, $\lambda^{(*,1)}$ is bigger than $\lambda^{(*,2)}$, $\lambda^{(*,3)}$; however, in subsequent iterations, the regularization parameter displays a consistent increase. The right frame of figure 15 displays a few selected $\mathcal{L}$-curves as computed at the final iteration of each inner iteration cycle, when the new weights are used; even if the $\mathcal{L}$-curves are clearly nested, the algorithm for detecting the corner selects points

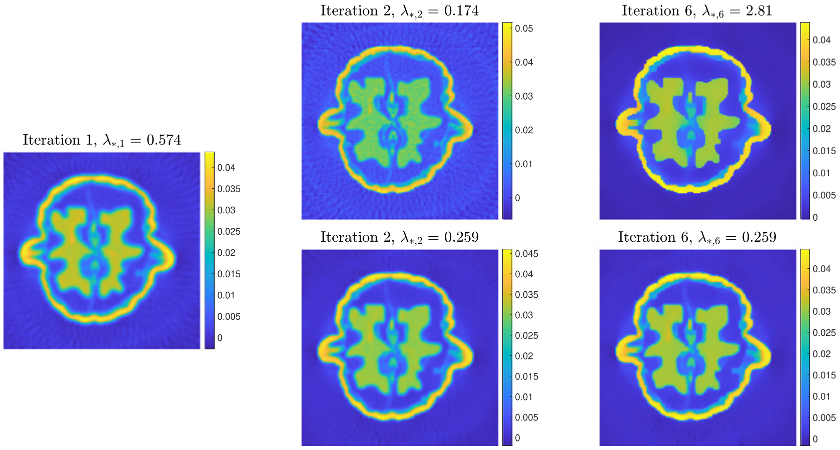
Iteration 1, $\lambda_{*,1} = 0.574$
Iteration 2, $\lambda_{*,2} = 0.174$
Iteration 6, $\lambda_{*,6} = 2.81$
Iteration 2, $\lambda_{*,2} = 0.259$
Iteration 6, $\lambda_{*,6} = 0.259$
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0

Figure 14. Undersampled fan-beam CT test problem. Reconstructions obtained at different outer iterations with the new and IRN–TV weights, displayed above and below, respectively ($x^{(*,1)}$ is computed without reweighting, so the two reconstructions coincide). The regularization parameters chosen by the hybrid method equipped with the $\mathcal{L}$-curve criterion are reported above each image.

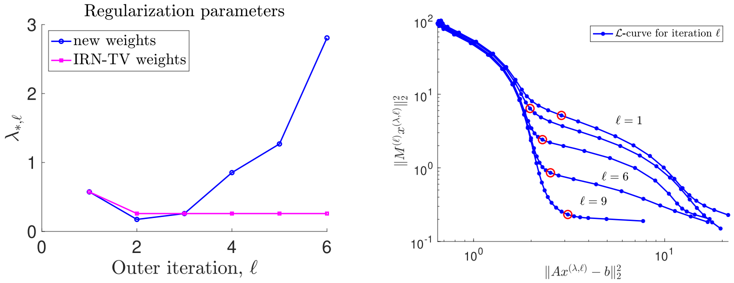

Figure 15. Undersampled fan-beam CT test problem. Left frame: regularization parameter versus number of (outer) iterations, selected according to the $\mathcal{L}$-curve criterion. Right frame: logarithmic plot of the $\mathcal{L}$-curves at the end of selected inner iterations, when the new weights are adopted (the red circles denote corners of each $\mathcal{L}$-curve, which corresponds to the chosen regularization parameter).

in the $\mathcal{L}$-curves that not necessarily correspond to regularization parameters that increase as ℓ increases.

5.2. Examples from image deblurring

Gaussian blur. We consider the exact sharp geometric image that contains piecewise constant objects of different sizes, as well as piecewise smooth objects. We corrupt it by applying a symmetric Gaussian blur of medium intensity (i.e., with variance 4) and adding 1% normally distributed (white Gaussian) noise. This test problem can be easily set up using

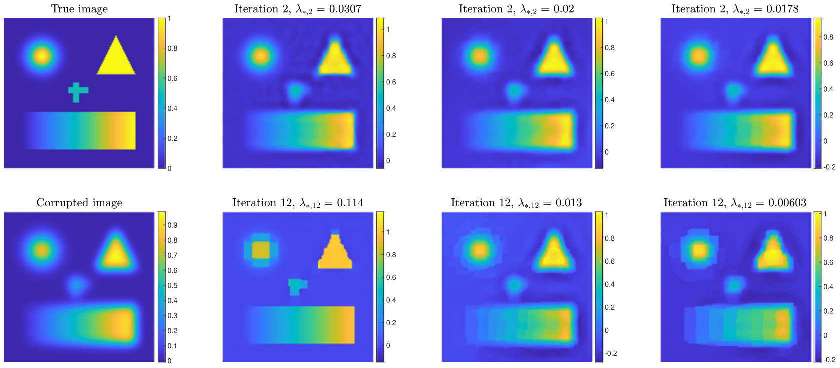
True image
1
0.8
0.6
0.4
0.2
0
Iteration 2, $\lambda_{*,2} = 0.0307$
1
0.8
0.6
0.4
0.2
0
Iteration 2, $\lambda_{*,2} = 0.02$
1
0.8
0.6
0.4
0.2
0
Iteration 2, $\lambda_{*,2} = 0.0178$
0.8
0.6
0.4
0.2
0
-0.2
Corrupted image
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Iteration 12, $\lambda_{*,12} = 0.114$
1
0.8
0.6
0.4
0.2
0
Iteration 12, $\lambda_{*,12} = 0.013$
1
0.8
0.6
0.4
0.2
0
-0.2
Iteration 12, $\lambda_{*,12} = 0.00603$
1
0.8
0.6
0.4
0.2
0
-0.2

Figure 16. Gaussian blur test problem. Left frames: true sharp image x (above) and corrupted image b (below). From left to right, the upper remaining frames display the initial reconstructions computed by the new method, the IRN–TV weights, and the IRN–aTV weights. From left to right, the lower remaining frames display the final reconstructions computed by the new method, the IRN–TV weights, and the IRN–aTV weights. Outer iteration counter, and regularization parameter obtained by the discrepancy principle are displayed above each frame.

IR tools. The leftmost frames of figure 16 show the true sharp image and the measured data b.

As in the previous examples, we run our algorithm using the discrepancy principle as parameter choice strategy within the inner hybrid scheme for generalized Tikhonov regularization. We compare the new solver with the analogous inner–outer scheme where the IRN–TV weights (5.1) and the IRN–aTV weights (5.2) are updated at each outer iteration. We prescribe 12 outer iterations to be performed; the stopping criterion described in section 4.3 (i.e., stop when $\|Lx^{(*,\ell)}\|_2$ decreases during two consecutive outer iterations) is never satisfied in this range of iterations. Figure 16 shows the reconstructions obtained for $\ell = 2$ (i.e., as soon as the new, IRN–TV and IRN–aTV weights become effective) and for $\ell = 12$ (i.e., when the iterations are stopped). Figure 17 reports the values of the relative errors and regularization parameter versus the number of iterations, for all the solvers.

Looking at figure 17 we can clearly see that, when applying the new method, the relative error greatly decreases during the early outer iterations, and then somewhat suffers from a slight increase. On the contrary, the relative error associated to the methods based on the IRN–TV and the IRN–aTV reweightings always increases as the outer iterations proceed; more precisely, for the IRN–TV weights a slight increase is already visible when the reweighting becomes active (i.e., at the second outer iteration), and a more substantial increase is noticeable at the third outer iteration, after which some stabilization (at roughly the same value computed by the IRN–aTV weights) happens without further improvements. Indeed, looking at the IRN–TV and the IRN–aTV reconstructions displayed in figure 16 we can see that, although some deblurring takes place, edges are not accurately recovered: specifically, in the last reconstruction, a lot of blocky artefacts are introduced, which look quite different in the isotropic and anisotropic cases; in the IRN–TV reconstructions not all the geometrical objects appearing in the true image are clearly distinguishable. The reconstruction computed by the new method has a somewhat blocky appearance, too; however edges, smaller objects, and

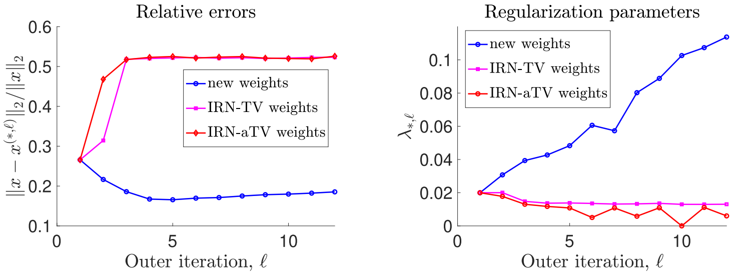

Figure 17. Gaussian blur test problem. Relative errors and regularization parameter versus number of (outer) iterations; all the methods use the hybrid method for general form Tikhonov during the inner iterations, and adaptively select the regularization parameter according to the discrepancy principle.

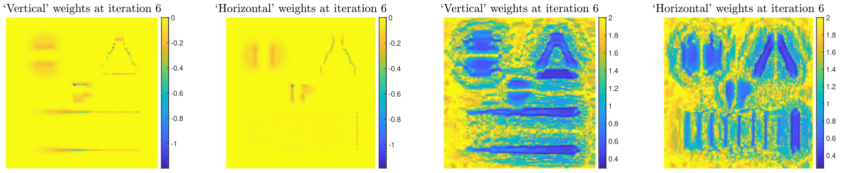

Figure 18. Gaussian blur test problem. First and second frame, from the left: new weights, to be applied to the vertical and the horizontal derivatives, respectively. Third and fourth frame, from the left: IRN–aTV weights, to be applied to the horizontal derivatives.

linearly varying pixel intensities can be recovered. To better understand the different behavior of the new and the IRN–aTV methods, the weights used at the 6th outer iteration of the two methods are displayed in figure 18, in logarithmic scale and reshaped as images (note that, since the IRN–aTV weights are different in the vertical and horizontal directions, they should be displayed in two different frames). We can clearly see that most of the edges of the objects in the original image show up in the new weights (when taking both the horizontal and vertical derivatives), and they are assigned a lower weight to discourage penalization during the reconstruction process. The IRN–aTV weights are not so effective in revealing the structure of the image, and some blocky artefacts are also visible within them. As explained in section 4, this behavior of the new weights greatly contributes to the success of the new algorithm. Finally we remark that, looking at the rightmost frame of figure 17, one can see that the regularization parameters selected by the discrepancy principle for the new method are not always increasing (indeed, there is a decrease at iteration 7): although this may seem to contradict the analysis performed in section 4.4, it should be taken into account that problem (4.1) (rather than problem (4.7)) is solved in practice, and therefore proposition 4.1 is not guaranteed to hold in practice.

Speckle blur. We use IR tools to define the following image deblurring simulation. Firstly, we would like to change the sharp image to the well-known *satellite* one, of size 256×256 pixels. Secondly, we would like to consider a so-called 'speckle' blur of mild intensity, which simulates a spatially invariant blur caused by atmospheric turbulence, and add 1% normally

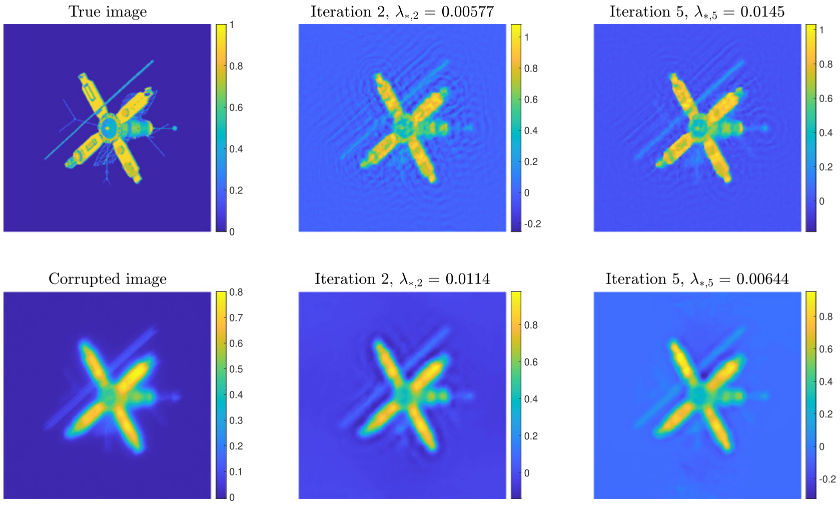

Figure 19. Speckle blur test problem. Left frames: true sharp image x (above) and corrupted image b (below). The upper remaining frames display the reconstructions computed by the new method; the lower remaining frames display the reconstructions computed by the IRN–TV weights. Outer iteration counter, and regularization parameter obtained by the discrepancy principle are displayed above each frame.

distributed (white Gaussian) noise. More precisely, we use the following MATLAB statements:

```
ProblemOptions = PRset(’trueImage’, ’satellite’, ’BlurLevel’, mild);
[A, b_true, x_true, ProblemInfo] = PRblurspeckle(256, ProblemOptions);
b = PRnoise(b_true, 0.01);
```

The leftmost frames of figure 19 show the true image and the measured data b.

As for the previous test problems, we assess the performance of the new solver by comparing it to the method that exploits the inner–outer iterative scheme based on the hybrid solver for general Tikhonov regularization together with the reweighting strategy based on the IRN–TV updating rule (5.1). Both algorithms adaptively select the regularization parameter during the inner iterations using the discrepancy principle. Only 5 outer iterations are performed, as the IRN–TV method progresses very slowly because, when the outer and inner iterations advance, a considerable effort is made by LSQR to compute the orthogonal projections that are needed to update the partial joint bidiagonalization (3.7) (see section 3.2 and [20] for more details). Indeed, after the first outer iteration, 1000 LSQR iterations (i.e., the maximum number of allowed LSQR iterations) are consistently performed for each inner iteration of the hybrid method.

Figure 20 shows a plot of the relative errors and chosen regularization parameters for the two methods at each outer iteration. Figure 19 shows the reconstructions obtained for $\ell = 2$ (i.e., as soon as the new and IRN–TV weights become effective) and for $\ell = 5$ (i.e., when the iterations are stopped). We can clearly see that the relative error decreases when the new weights are employed, but displays the opposite trend when the IRN–TV weights are employed. Looking at the images in figure 19 we realise that, similarly to the previous example, the IRN–TV

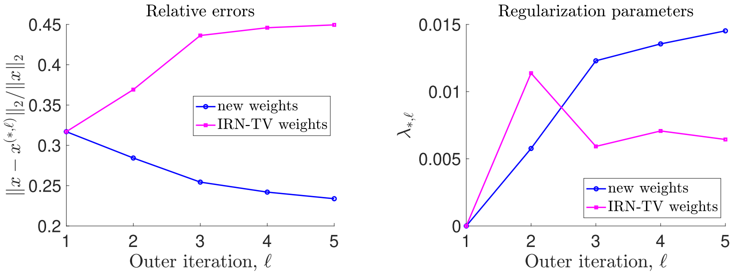

Figure 20. Speckle blur test problem. Relative errors and regularization parameters values at each outer iteration ℓ. The regularization parameter is computed by applying the discrepancy principle at each iteration of the hybrid method; 10^{-8} is set as the minimum value of the regularization parameter for the discrepancy principle.

weights have the tendency of over-smoothing the solution, resulting in blocky images where many edges are not clearly visible (leading to an increase in the relative error). On the contrary, the displayed reconstructions obtained by the new method look slightly undersmoothed and some ringing artefacts are visible; however, such artefacts may be removed if more outer iterations are performed; also, pixel values and edges are more accurately recovered when the new weights are employed.

6. Conclusions and future work

In this paper we introduced a new inner–outer iterative algorithm for restoring and reconstructing images with enhanced edges, where a sequence of quadratic Tikhonov-regularized problems is solved (outer iterations) and a specific hybrid method is applied to solve each quadratic problem in the sequence (inner iterations). The regularization matrix for each quadratic problem in the sequence is updated at each outer iteration and is obtained by premultiplying the gradient operator by a weight matrix that encodes edge information disclosed within all the previous outer iterations, in such a way that edges are gradually enhanced in the solution process. The hybrid inner solver is very effective, and well-known strategies to set the regularization parameter can be successfully incorporated therein. The resulting strategy is innovative and very efficient when compared to available edge-recovery techniques, mainly because of the definition of the new weights and the incorporation of automatic regularization parameter choice techniques.

Future work will include the investigation of similar reweighting techniques, where the inner iterations are performed by a different hybrid method (for instance the one described in section 3.1, with possible strategies to approximate the action of $M_A^\dagger$). Also, different expressions of the weighting matrices may be considered, in such a way that edge information is extracted by operators other than a rescaled gradient; indeed, the current weights and the overall quality of the reconstruction may be unfavorably affected by possible spurious edges recovered during the iterative process. Finally, similarly to [7, 11, 12], possible hybrid variants that involve the use of flexible Krylov subspaces will be investigated, with the ideal goal of avoiding nested cycles of iterations.

ORCID iDs

Silvia Gazzola https://orcid.org/0000-0001-9588-0896

James G Nagy https://orcid.org/0000-0002-0451-3853

Eric L Miller https://orcid.org/0000-0002-3156-6002

References

[1] Arridge S R, Betcke M M and Harhanen L 2014 Iterated preconditioned LSQR method for inverse problems on unstructured grids *Inverse Problems* **30** 075009

[2] Bardsley J 2012 Laplace-distributed increments, the laplace prior, and edge-preserving regularization *J. Inverse Ill-Posed Problems* **20** 271–85

[3] Björck Å 1988 A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations *BIT Numer. Math.* **28** 659–70

[4] Björck Å, Grimme E and van Dooren P 1994 An implicit shift bidiagonalization algorithm for ill-posed systems *BIT Numer. Math.* **34** 510–34

[5] Chan T and Shen J 2005 *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods* (Philadelphia, PA: SIAM)

[6] Chen D, MacLachlan S and Kilmer M 2011 Iterative parameter-choice and multigrid methods for anisotropic diffusion denoising *SIAM J. Sci. Comput.* **33** 2972–94

[7] Chung J and Gazzola S 2019 Flexible Krylov methods for ℓ_p regularization *SIAM J. Sci. Comput.* **41** S149–71

[8] Chung J M, Kilmer M E and O'Leary D P 2015 A framework for regularization via operator approximation *SIAM J. Sci. Comput.* **37** B332–S580

[9] Fong D C L and Saunders M 2013 LSMR: an iterative algorithm for sparse least-squares problems *SIAM J. Sci. Comput.* **33** 2950–71

[10] Gazzola S, Hansen P C and Nagy J G 2019 IR tools: a MATLAB package of iterative regularization methods and large-scale test problems *Numer. Algorithms* **81** 773–811

[11] Gazzola S and Sabaté Landman M 2019 Flexible GMRES for total variation regularization *BIT Numer. Math.* **59** 721–46

[12] Silvia Gazzola and James G Nagy 2014 Generalized Arnoldi–Tikhonov method for sparse reconstruction *SIAM J. Sci. Comput.* **36** B225–47

[13] Gazzola S, Novati P and Russo M R 2015 On Krylov projection methods and Tikhonov regularization *Electron. Trans. Numer. Anal.* **44** 83–123

[14] Golub G H and Loan C F V 1996 *Matrix Computations* 3rd edn (Baltimore, MD: John Hopkins University Press)

[15] Gorodnitsky I F and Rao B D 1992 A new iterative weighted norm minimization algorithm and its applications *IEEE 6th SP Workshop on Statistical Signal and Array Processing* pp 412–5

[16] Hämäläinen K, Harhanen L, Kallonen A, Kujanpää A, Niemi E and Siltanen S 2015 Tomographic x-ray data of walnut (arXiv:1502.04064)

[17] Hansen P C 1997 *Rank-deficient and Discrete Ill-Posed Problems* (Philadelphia, PA: SIAM)

[18] Hansen P C and Jørgensen J S 2018 AIR tools II: algebraic iterative reconstruction methods, improved implementation *Numer. Algorithms* **79** 107–37

[19] Hansen P C, Nagy J G and O'Leary D P 2006 *Deblurring Images: Matrices, Spectra and Filtering* (Philadelphia, PA: SIAM)

[20] Kilmer M E, Hansen P C and Español M I 2007 A projection-based approach to general-form Tikhonov regularization *SIAM J. Sci. Comput.* **29** 315–30

[21] Kilmer M E and O'Leary D P 2001 Choosing regularization parameters in iterative methods for ill-posed problems *SIAM J. Matrix Anal. Appl.* **22** 1204–21

[22] Li Y and Santosa F 1996 A computational algorithm for minimizing total variation in image restoration *IEEE Trans. Image Process.* **5** 987–95

[23] O'Leary D P and Simmons J A 1981 A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems *SIAM J. Sci. Comput.* **2** 474–89

[24] Osher S, Burger M, Goldfarb D, Xu J and Yin W 2005 An iterative regularization method for total variation-based image restoration *Multiscale Model. Simul.* **4** 460–89

[25] Paige C C and Saunders M A 1982 LSQR: an algorithm for sparse linear equations and sparse least squares *ACM Trans. Math. Softw.* **8** 43–71

[26] Paige C C and Saunders M A 1982 Algorithm 583: LSQR: sparse linear equations and least-squares problems *ACM Trans. Math. Softw.* **8** 195–209

[27] Reichel L and Shyshkov A 2008 A new zero-finder for Tikhonov regularization *BIT Numer. Math.* **48** 627–43

[28] Renaut R A, Vatankhah S and Ardestani V E 2017 Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems *SIAM J. Sci. Comput.* **39** B221–43

[29] Rodriguez P and Wohlberg B 2008 An efficient algorithm for sparse representations with ℓ^p data fidelity term *Proc. of 4th IEEE Andean Technical Conf.* (ANDESCON)

[30] Rudin L I, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Physica* D **60** 259–68

[31] Semerci O 2012 Image formation methods for dual energy and multi-energy computed tomography *PhD Thesis* Tufts University, Medford, MA

[32] Vogel C R 2002 *Computational Inverse Problems* (Philadelphia, PA: SIAM)

[33] Vogel C R and Oman M E 1996 Iterative methods for total variation denoising *SIAM J. Sci. Comput.* **17** 227–38

[34] Wohlberg B and Rodríguez P 2007 An iteratively reweighted norm algorithm for minimization of total variation functionals *IEEE Signal Process. Lett.* **14** 948–51