**J Math Imaging Vis manuscript No.**
(will be inserted by the editor)

# On the convergence of primal–dual hybrid gradient algorithms for total variation image restoration

**Silvia Bonettini** · **Valeria Ruggiero**

Received: date / Accepted: date

**Abstract** In this paper we establish the convergence of a general primal–dual method for nonsmooth convex optimization problems whose structure is typical in the imaging framework, as, for example, in the Total Variation image restoration problems. When the steplength parameters are *a priori* selected sequences, the convergence of the scheme is proved by showing that it can be considered as an $\varepsilon$-subgradient method on the primal formulation of the variational problem. Our scheme includes as special case the method recently proposed by Zhu and Chan for Total Variation image restoration from data degraded by Gaussian noise. Furthermore, the convergence hypotheses enable us to apply the same scheme also to other restoration problems, as the denoising and deblurring of images corrupted by Poisson noise, where the data fidelity function is defined as the generalized Kullback–Leibler divergence or the edge preserving removal of impulse noise. The numerical experience shows that the proposed scheme with a suitable choice of the steplength sequences performs well with respect to state-of-the-art methods, especially for Poisson denoising problems, and it exhibits fast initial and asymptotic convergence.

**Keywords** Convex optimization · Primal–Dual Hybrid Gradient method · Total Variation · $\varepsilon$–subgradient method · Kullback–Leibler divergence.

This work is supported by the PRIN2008 Project of the Italian Ministry of University and Research, grant 2008T5KA4L, *optimizAtion Methods and Software for Inverse Problems, http://www.unife.it/prisma.*

S. Bonettini
Department of Mathematics, University of Ferrara
Tel.: +39-0532-974785
E-mail: silvia.bonettini@unife.it

V. Ruggiero
Department of Mathematics and LTTA Lab, University of Ferrara
Tel.: +39-0532-974783
E-mail: valeria.ruggiero@unife.it

## 1 Introduction

Image restoration is an inverse problem that consists in finding an approximation of the original object $\tilde{x} \in R^n$ from a set $g \in R^m$ of detected data. In a discrete framework, we assume that each component of the data $g_i$ is the realization of a random variable whose mean is $(H\tilde{x}+b)_i$, where $H \in R^{m\times n}$ represents the distortion due to the acquisition system and $b \in \mathbb{R}^m$ is a nonnegative constant backgroung term. We assume that $H$ is known and, in particular, when $H = I$ we have a denoising problem while, in the other cases, we deal with a deblurring problem. In the Bayesian framework [27,17], an approximation of the original object $\tilde{x}$ is obtained by solving a minimization problem where the objective function is the combination of two terms: the first one is a nonnegative functional measuring the data discrepancy, to be chosen according to the noise statistics, while the second one is a regularization term weighted by a positive parameter balancing the two terms. Some physical constraint can be added, such as non negativity and flux conservation. When the goal is preserving sharp discontinuities while removing noise and blur, we can use as regularization term the Total Variation (TV) functional (introduced first in [25]), which, in the discrete framework, is defined as

$$\sum_{k=1}^{n} \|(\nabla x)_k\| \tag{1}$$

where $(\nabla x)_k$ denotes a discrete approximation of the gradient of $x$ at the pixel $k$.
For Gaussian noise, the fit–to–data term is given by the following quadratic function

$$\frac{1}{2}\|Hx+b-g\|^2 \tag{2}$$

and the variational model combining (2) and (1) has been extensively studied; its primal and dual formulations have been

deeply investigated in order to design algorithms specifically tailored for image processing applications. The large size of these problems requires schemes with a low computational cost per iteration (typically only a matrix-vector product per iteration) and a fast initial convergence that enables to obtain medium accurate and visually satisfactory results in a short time. In the class of first order methods, requiring only function and gradient evaluations, popular methods for TV minimization include the time-marching method [25], split Bregman [22,19], Chambolle's [8,9,4], gradient projection [32,29], Nesterov-type methods [14,2]. For the most part of these algorithms, Matlab codes are available in the public domain (see [14] for references).

We mention also second-order methods proposed in [11,18]. These can be quadratically convergent, but they require the solution of a system at each iteration and information about the Hessian matrix.

Another approach is based on the primal–dual formulation of (3) as saddle point problem. In [31], Zhu and Chan propose a first–order method named Primal–Dual Hybrid Gradient (PDHG) method, where at any iteration both primal and dual variables are updated by descent and ascent gradient projection steps respectively. Furthermore, the authors propose to let the steplength parameters varying trough the iterations and to choose them as prefixed sequences. The twofold aim is to avoid the difficulties that arise when working only on the primal or dual formulation and to obtain a very efficient scheme, well suited also for large scale problems.

The convergence of PDHG has been investigated in [16], where the algorithm with variable stepsizes is interpreted as a projected averaged gradient method on the dual formulation, while in [10] the convergence of PDHG with fixed stepsizes is discussed (see also [2]). Numerical experiments [31,16,10] show that the method exhibits fast convergence for some special choices of the steplength sequences, but, at the best of our knowledge, a theoretical justification of the convergence of the PDHG scheme with these *a priori* choices is still missing.

A recent development on the primal–dual methods can be found also in [10] as a special case of a primal–dual algorithm for the minimization of a convex relaxation of the Mumford-Shah functional. This last algorithm generalizes the classical Arrow-Hurwicz algorithm [1] and converges for constant steplength parameters. Furthermore, for uniformly convex objective functions, a convenient strategy to devise adaptive step sizes is theoretically obtained and it seems to be numerically effective.

The aim of this paper is to define a robust convergence framework for primal–dual methods with variable steplength parameters which apply to optimization problems of the form

$$\min_{x\in X} f_0(x) + f_1(Ax) \tag{3}$$

where $f_0$ and $f_1$ are convex proper lower semicontinuous functions, over $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively, not necessarily differentiable, $A \in \mathbb{R}^{m\times n}$ and $X$ represents the domain of the objective function or a subset of it expressing physical features. The key point of our analysis is to consider the primal–dual method as an $\varepsilon$–subgradient method [13,23,20] on the primal formulation (3). Then, the two main contributions of this paper are the following:

- we establish the convergence proof of a primal–dual method where the steplength are chosen as *a priori* sequences; this analysis provides also, as a special case, the convergenge proof of the PDHG method [31] with the steplength choices suggested by the authors as the best performing one on the variational problem (2)–(1);
- we design a general algorithm which can be applied to general TV restoration problems such as
  - the denoising or deblurring of images corrupted by Poisson noise, where the data discrepancy is expressed by the generalized Kullback–Leibler (KL) divergence:

$$f_0(x) = \sum_k \left\{ g_k \ln \frac{g_k}{(Hx+b)_k} + (Hx+b)_k - g_k \right\} \tag{4}$$

    with $g_k \ln g_k = 0$ if $g_k = 0$;
  - the edge preserving removal of impulse noise, where a suitable fit-to-data term is the $\ell_1$ norm

$$f_0(x) = \|x-g\|_1$$

The paper is organized as follows: in Section 2 some basic definitions and results about $\varepsilon$-subgradient and $\varepsilon$-subgradient methods are restated. In Section 3 we introduce the primal–dual scheme and the connections with $\varepsilon$-subgradient methods are investigated. In particular, we provide the convergence analysis for primal–explicit and primal–implicit schemes. In Section 4 some applications of our results are described. The numerical experiments in section 5 show that the proposed scheme, with a suitable choice of the steplength sequences, can be a very effective tool for TV image restoration also in presence of Poisson and impulse noise.

## 2 Definitions and preliminary results

We denote by $\mathbb{R}^n$ the usual $n$-dimensional Euclidean space, by $\langle x,y\rangle = x^T y$ the inner product of two vectors of $\mathbb{R}^n$ and by $\|\cdot\|$ the $l_2$ norm.

The domain of a function $f : \mathbb{R}^n \to ]-\infty,+\infty]$ is $\mathrm{dom}(f) = \{x \in \mathbb{R}^n : f(x) < +\infty\}$. A function $f$ is said *proper* if $\mathrm{dom}(f) \neq \emptyset$. The diameter of a set $X$ is defined as $\mathrm{diam}(X) = \max_{x,z\in X} \|x-z\|$.

Let $P_\Omega(z)$ denote the orthogonal projection of the point $z \in \mathbb{R}^n$ onto the nonempty, close, convex set $\Omega \subseteq \mathbb{R}^n$, $P_\Omega(z) = \arg\min_{u\in\Omega} \frac{1}{2}\|u-z\|^2$.

We recall that for a convex function $f$, the resolvent operator $(I+\theta\partial f)^{-1}$ is defined as

$$(I+\theta\partial f)^{-1}(z) = \arg\min f(x) + \frac{1}{2\theta}\|x-z\|^2$$

where $\partial f$ is the subdifferential mapping and $\theta$ is a positive parameter.

**Definition 1** *[24, §23]. Let $f$ a proper convex function on $\mathbb{R}^n$.*
*The $\varepsilon$-subdifferential of $f$ at $x \in dom(f)$, defined for $\varepsilon \in \mathbb{R}$, $\varepsilon \geq 0$, is the set*

$$\partial_\varepsilon f(x) = \{w \in \mathbb{R}^n : \ f(z) \geq f(x) + \langle w, z-x\rangle - \varepsilon, \ \ \forall z \in \mathbb{R}^n\}$$

For $\varepsilon = 0$ the definition of subdifferential is recovered while for $\varepsilon > 0$ we have a larger set; furthermore, for $\varepsilon_1 > \varepsilon_2 > 0$, we have $\partial_{\varepsilon_1} f(x) \supseteq \partial_{\varepsilon_2} f(x) \supseteq \partial f(x)$. Every element of $\partial_\varepsilon f(x)$ is an *$\varepsilon$-subgradient* of $f$ at $x$. In the following we will make use of the linearity property of the $\varepsilon$-subgradient, which, for sake of completeness, is restated below.

**Property 1** *If $f(x) = \sum_{i=1}^n \alpha_i f_i(x)$, where $\alpha_i \geq 0$, $w_i \in \partial_{\varepsilon_i} f_i(x)$ and $x \in \bigcap_{i=1}^n dom(f_i)$, then $\sum_{i=1}^n \alpha_i w_i \in \partial_\varepsilon f(x)$, where $\varepsilon = \sum_{i=1}^n \varepsilon_i$.*

*Proof.* By Definition 1, we have

$$f_i(z) - f_i(x) \geq \langle w_i, z-x\rangle - \varepsilon_i \quad i = 1,...,n$$

Then, the claim follows by multiplying the previous inequalities by $\alpha_i$ and summing up for $i = 1,...,n$. □

**Definition 2** *[24, §12]. The conjugate of a convex function $f$ is the function $f^*$ defined by*

$$f^*(y) = \sup_x \langle x, y\rangle - f(x)$$

If $f(x)$ is lower semicontinuous and proper, then $f^*$ is lower semicontinuous and $f^{**} = f$.

**Proposition 1** *Let $f(x)$ a proper lower semicontinuous convex function. Then, for every $x \in dom(f)$ and $y \in dom(f^*)$ we have $y \in \partial_\varepsilon f(x)$, with $\varepsilon = f(x) - (\langle y, x\rangle - f^*(y))$.*

*Proof.* Let $x, z \in \mathbb{R}^n$ and $y \in \mathrm{dom}(f^*)$. Then, we can write

$$\begin{aligned} f(x) + \langle y, z-x\rangle &= f(x) - (\langle y,x\rangle - f^*(y)) + \langle y, z\rangle - f^*(y) \\ &\leq f(x) - (\langle y,x\rangle - f^*(y)) + \sup_y \langle y, z\rangle - f^*(y) \\ &= \underbrace{f(x) - (\langle y,x\rangle - f^*(y))}_{=\varepsilon} + f(z) \end{aligned}$$

Since $f(x) = \sup_x \langle y, x\rangle - f^*(y)$, then $\varepsilon \geq 0$; thus, Definition 1 is fulfilled. □

We remark that $\varepsilon = 0$ (that is $y \in \partial f(x)$) if and only if $f(x) = \langle y, x\rangle - f^*(y)$. Now we state the following corollary, which is useful for the subsequent analysis; its proof can be carried out by employing similar arguments as in the previous proposition.

**Corollary 1** *Let $f$ a proper lower semicontinuous convex function and let $A$ be a linear operator. Consider the composition $(f \circ A)(x) = f(Ax)$: then, for every $x \in dom(f \circ A)$ and $y \in dom(f^*)$ we have $A^T y \in \partial_\varepsilon (f \circ A)(x)$, with $\varepsilon = f(Ax) - (\langle y, Ax\rangle - f^*(y))$.*

An important property of the $\varepsilon$-subgradients is their boundedness over compactly contained subsets of int $\mathrm{dom}(f)$, as we prove in the following proposition.

**Proposition 2** *Assume that $S$ is a compactly contained bounded subset of int $dom(f)$. Then, the set $\bigcup_{x\in S} \partial_\varepsilon f(x)$ is nonempty, closed and bounded.*

*Proof.* Let $\lambda > 0$ be such that $S + B^\lambda \subseteq \mathrm{int\ dom}(f)$, where $B^\lambda$ is the ball of $\mathbb{R}^n$ with radius $\lambda$ and $S + B^\lambda = \{u \in \mathbb{R}^n : \|u - x\| \leq \lambda, x \in S\}$. By Theorem 6.2 in [15] it follows that $\bigcup_{x\in S} \partial_\varepsilon f(x) \subseteq \bigcup_{x\in S+B^\lambda} \partial f(x) + B^{\frac{\varepsilon}{\lambda}}$. The last term in the previous inclusion is nonempty, closed and bounded (see [24, Theorem 24.7]); thus the theorem follows. □

**Proposition 3** *Let $x, \bar{x} \in dom f$ and $g \in \partial f(x)$; then, $g \in \partial_\varepsilon f(\bar{x})$ with $\varepsilon = D_f(\bar{x}, x)$, where $D_f(\bar{x}, x) = f(\bar{x}) - f(x) - \langle g, \bar{x} - x\rangle$ is the Bregman divergence associated with $f$ at $x$.*

*Proof.* Since $g \in \partial f(x)$, for all $z$ we have

$$\begin{aligned} f(z) &\geq f(x) + \langle g, z-x\rangle \\ &= f(\bar{x}) + \langle g, z-\bar{x}\rangle - \underbrace{(f(\bar{x}) - f(x) - \langle g, \bar{x}-x\rangle)}_{\varepsilon} \end{aligned}$$

where $\varepsilon \geq 0$ by the hypothesis on $g$. □

### 2.1 The $\varepsilon$-subgradient projection method

Consider the constrained minimization problem

$$\min_{x\in X} f(x), \tag{5}$$

where $f$ is a convex, proper, lower semicontinuous function; the $\varepsilon$-subgradient projection method is defined as follows

$$x^{(k+1)} = P_X(x^{(k)} - \theta_k w^{(k)}), \ \ w^{(k)} \in \partial_{\varepsilon^k} f(x^{(k)}) \tag{6}$$

given the steplength sequence $\{\theta_k\}$ and subgradient residuals $\{\varepsilon^k\}$ (see for example [13, 23, 20] and reference therein). The convergence properties of a subgradient method strongly depends on the steplength choice, and different selection strategies can be devised in the literature (see [5, Chapter

6.8] for a recent review). In this paper we focus on the *diminishing divergent series stepsize rule*, that consists in choosing any sequence of positive steplength $\theta_k > 0$ such that

A1 $\lim_{k\to\infty} \theta_k = 0$

A2 $\sum_{k=0}^{\infty} \theta_k = \infty$.

The convergence of the $\varepsilon$-subgradient projection method can be stated as in [20, Theorem 3]. For sake of completeness we report the statement below.

**Theorem 1** *Let $\{x^{(k)}\}$ be the sequence generated by the method* (6) *and assume that the set $X^*$ of the solutions of* (5) *is bounded. Under the assumptions A1–A2, if $w^{(k)}$ is bounded and $\lim_k \varepsilon^k = 0$, then $\{f(x^{(k)})\}$ converges to a minimum of $f(x)$ over $X$ and dist$(x^{(k)}, X^*) \to 0$.*

**Remark**. When problem (5) has a unique solution $x^*$, the previous Theorem assures the convergence of the sequence $\{x^{(k)}\}$ to the minimum point $x^*$.

## 3 The primal-dual scheme

We consider the minimization of

$$f(x) \equiv f_0(x) + f_1(Ax)$$

where $f_0(x)$ and $f_1(x)$ are convex, proper, lower semicontinuous functions, not necessarily differentiable, and such that

$$\text{diam}(\text{dom}(f_1^*)) = D < +\infty \tag{7}$$

We consider problems of the form (5) where the constraint set $X$ is a closed convex subset $X$ of dom$(f)$; otherwise, for the unconstrained case, we set $X = \text{dom}(f)$. We assume also that the set of the minimum points $X^*$ is bounded.
We remark that, the boundedness of $X^*$ together with (7), ensures that the min–max theorem [24, p.397] holds. As a consequence of this, the minimization problem (5) is equivalent to the saddle point problem

$$\min_{x\in X} \max_{y} F(x,y) \equiv f_0(x) + \langle y, Ax\rangle - f_1^*(y) \tag{8}$$

Indeed, by definition, the pair $(x^*, y^*)$ is a saddle point of (8) when

$$F(x^*, y) \le F(x^*, y^*) \le F(x, y^*) \ \ \forall x \in X \tag{9}$$

and (9) holds if and only if $A^T y^* \in \partial f_1(x^*)$ and $x^* \in X^*$.

### 3.1 Primal–explicit scheme

We consider the following algorithm.

**Algorithm 1**

Choose the starting point $x^{(0)} \in X$, $y^{(0)} \in \text{dom}(f_1^*)$
FOR $k = 0, 1, 2, \ldots$ DO THE FOLLOWING STEPS:

STEP 1. Choose positive steplength parameters $\tau_k$ and $\theta_k$;
STEP 2. Compute

$$y^{(k+1)} \leftarrow (I + \tau_k \partial f_1^*)^{-1}(y^{(k)} + \tau_k A x^{(k)}) \tag{10}$$

$$x^{(k+1)} \leftarrow P_X(x^{(k)} - \theta_k(g^{(k)} + A^T y^{(k+1)})) \tag{11}$$

with $g^{(k)} \in \partial_{\delta_k} f_0(x^{(k)})$, $\delta_k \ge 0$

STEP 3. Terminate if a stopping criterion is satisfied: otherwise, go to step 1.

END

In order to prove the convergence of Algorithm 1, we make the following assumptions on the parameters $\tau_k$ and $\delta_k$:

A3 $\lim_{k\to\infty} \tau_k = \infty$;

A4 $\lim_{k\to\infty} \delta_k = 0$;

**Lemma 1** *Let $x^{(k)}$ any sequence in dom$(f)$ and assume that* (7) *holds. Under the assumptions A1 and A3, we have that $A^T y^{(k+1)} \in \partial_{\varepsilon^k}(f_1 \circ A)(x^{(k)})$ with $\lim_{k\to\infty} \varepsilon^k = 0$.*

*Proof.* Corollary 1 guarantees that $A^T y^{(k+1)} \in \partial_{\varepsilon^k}(f_1 \circ A)(x^{(k)})$ with $\varepsilon^k = f_1(Ax^{(k)}) + f_1^*(y^{(k+1)}) - \langle y^{(k+1)}, Ax^{(k)}\rangle$.
By (10) it follows that

$$\begin{aligned} y^{(k+1)} &= \arg\min_y f_1^*(y) + \frac{1}{2\tau_k}\|y - (y^{(k)} + \tau_k A x^{(k)})\|^2 \\ &= \arg\min_y f_1^*(y) - \langle y, Ax^{(k)}\rangle + \frac{1}{2\tau_k}\|y - y^{(k)}\|^2 \end{aligned}$$

We have

$$\begin{aligned} & f_1^*(y^{(k+1)}) - \langle y^{(k+1)}, Ax^{(k)}\rangle \\ & \le f_1^*(y^{(k+1)}) - \langle y^{(k+1)}, Ax^{(k)}\rangle + \frac{1}{2\tau_k}\|y^{(k+1)} - y^{(k)}\|^2 \\ & = \min_y f_1^*(y) - \langle y, Ax^{(k)}\rangle + \frac{1}{2\tau_k}\|y - y^{(k)}\|^2 \\ & \le \min_y f_1^*(y) - \langle y, Ax^{(k)}\rangle + \frac{1}{2\tau_k} D^2 \\ & \le -f_1(Ax^{(k)}) + \frac{1}{2\tau_k} D^2 \end{aligned}$$

which results in

$$\begin{aligned} & \varepsilon^k = f_1^*(y^{(k+1)}) - \langle y^{(k+1)}, Ax^{(k)}\rangle + f_1(Ax^{(k)}) \\ & \le \frac{1}{2\tau_k} D^2 \end{aligned}$$

Since $\tau_k \to +\infty$ and $\varepsilon^k \ge 0\ \forall k$, then $\varepsilon^k \to 0$. □

We are now ready to state the following convergence result

for Algorithm 1, which directly follows from Theorem 1 and Lemma 1.

**Theorem 2** *We assume $X^*$ bounded. Let $\{x^{(k)}\}$ be the sequence generated by Algorithm 1. Under the assumptions A1–A4, if $\{g^{(k)}\}$ is bounded, then $\{f(x^{(k)})\}$ converges to a minimum of $f(x)$ over $X$ and $dist(x^{(k)},X^*) \to 0$.*

One of the main advantages of Algorithm 1 is its generality, and different algorithms can be defined according to the strategy chosen to compute the approximate subgradient $g^{(k)}$.
In particular, when the function $f_0(x)$ is differentiable, we may set

$$g^{(k)} = \nabla f_0(x^{(k)}),$$

so that we have $\delta_k = 0$ for all $k$.

### 3.2 Primal–implicit scheme

We observe that the constrained problem (5) can be formulated also as

$$\min_{x\in\mathbb{R}^n} f_0^X(x) + f_1(Ax)$$

where $f_0^X(x) = f_0(x) + \iota_X(x)$ and $\iota_X(x)$ is the indicator function of the set $X$, defined as

$$\iota_X(x) = \begin{cases} 0 & \text{if } x \in X \\ +\infty & \text{if } x \notin X \end{cases}$$

We consider the following implicit version of Algorithm 1.

**Algorithm 2**

Choose the starting point $x^{(0)} \in X$, $y^{(0)} \in \mathrm{dom}(f_1^*)$
FOR $k = 0,1,2,\ldots$ DO THE FOLLOWING STEPS:

STEP 1. Choose positive steplength parameters $\tau_k$ and $\theta_k$;
STEP 2. Compute
$$y^{(k+1)} \leftarrow (I + \tau_k \partial f_1^*)^{-1}(y^{(k)} + \tau_k A x^{(k)}) \quad (12)$$
$$x^{(k+1)} \leftarrow (I + \theta_k \partial f_0^X)^{-1}(x^{(k)} - \theta_k A^T y^{(k+1)}) \quad (13)$$
STEP 3. Terminate if a stopping criterion is satisfied: otherwise, go to step 1.

END

Then, the new point $x^{(k+1)}$ is defined also as

$$x^{(k+1)} = \arg\min_{x\in X} f_0(x) + \frac{1}{2\theta_k}\|x - x^{(k)} + \theta_k A^T y^{(k+1)}\|^2$$

and, thus, the updating step (13) implies that

$$\frac{x^{(k)} - x^{(k+1)}}{\theta_k} - A^T y^{(k+1)} \in \partial f_0^X(x^{(k+1)}) \quad (14)$$

In particular, we have that $x^{(k+1)} \in \mathrm{dom}(f_0^X)$ and

$$x^{(k+1)} = x^{(k)} - \theta_k(g^{(k)} + A^T y^{(k+1)}) \quad (15)$$

where $g^{(k)} \in \partial f_0^X(x^{(k+1)})$. From Proposition 3, it follows that

$$g^{(k)} \in \partial_{\delta_k} f_0^X(x^{(k)})$$

with

$$\delta^k = f_0^X(x^{(k)}) - f_0^X(x^{(k+1)}) - \langle g^{(k)}, x^{(k)} - x^{(k+1)}\rangle \quad (16)$$

Thus, Algorithm 2 can be considered an $\varepsilon$-subgradient method and the convergence analysis previously developed still applies, as we show in the following. However, this semi–implicit version has stronger convergence properties: in particular, we are able to prove the boundedness of the iterates $\{x^{(k)}\}$.

**Lemma 2** *Assume that*

$$\sum_{k=0}^{\infty} \frac{\theta_k}{\tau_k} < +\infty \text{ and } \sum_{k=0}^{\infty} \theta_k^2 < +\infty \quad (17)$$

*Then, the sequence $\{x^{(k)}\}$ generated by Algorithm 2 is bounded.*

*Proof.* Let $(x^*, y^*)$ a saddle point of (8). From (14) and from the definition of the subgradient, we obtain

$$f_0^X(x^*) \geq f_0^X(x^{(k+1)}) + \frac{1}{\theta_k}\langle x^{(k)} - x^{(k+1)}, x^* - x^{(k+1)}\rangle + \\ -\langle A(x^* - x^{(k+1)}), y^{(k+1)}\rangle$$

where

$$\langle x^{(k)} - x^{(k+1)}, x^* - x^{(k+1)}\rangle = \\ = \frac{\|x^{(k+1)} - x^*\|^2}{2} + \frac{\|x^{(k+1)} - x^{(k)}\|^2}{2} - \frac{\|x^{(k)} - x^*\|^2}{2}$$

which gives

$$\frac{\|x^{(k+1)} - x^*\|^2}{2\theta_k} \leq \frac{\|x^{(k)} - x^*\|^2}{2\theta_k} - \frac{\|x^{(k+1)} - x^{(k)}\|^2}{2\theta_k} + \\ + f_0^X(x^*) - f_0^X(x^{(k+1)}) + \langle A^T y^{(k+1)}, x^* - x^{(k+1)}\rangle \quad (18)$$

Similarly, from (13) we obtain

$$\frac{\|y^{(k+1)} - y^*\|^2}{2\tau_k} \leq \frac{\|y^{(k)} - y^*\|^2}{2\tau_k} - \frac{\|y^{(k+1)} - y^{(k)}\|^2}{2\tau_k} + \\ + f_1^*(y^*) - f_1^*(y^{(k+1)}) + \langle Ax^{(k)}, y^{(k+1)} - y^*\rangle \quad (19)$$

We observe that, if we define $F^X(x,y) = f_0^X(x) + \langle x, A^T y\rangle - f_1^*(y)$, we have

$$f_1^*(y^*) - f_1^*(y^{(k+1)}) + \langle Ax^{(k)}, y^{(k+1)} - y^*\rangle + \\ + f_0^X(x^*) - f_0^X(x^{(k+1)}) + \langle A^T y^{(k+1)}, x^* - x^{(k+1)}\rangle = \\ = F^X(x^*, y^{(k+1)}) - F^X(x^{(k+1)}, y^*) + \\ + \langle A(x^{(k)} - x^{(k+1)}), y^{(k+1)} - y^*\rangle \leq \\ \leq \langle x^{(k)} - x^{(k+1)}, A^T(y^{(k+1)} - y^*)\rangle$$

where the last inequality follows from the saddle point property (9) of the pair $(x^*, y^*)$.
The last term can be further estimated as in [10, §3.2], by observing that for any $t \in (0,1]$ we have

$$\left\| \sqrt{\frac{t}{\theta_k}}(x^{(k+1)} - x^{(k)}) + \sqrt{\frac{\theta_k}{t}} A^T (y^{(k+1)} - y^*) \right\|^2 \geq 0$$

which yields

$$\langle x^{(k)} - x^{(k+1)}, A^T(y^{(k+1)} - y^*) \rangle \leq$$

$$\frac{t}{2\theta_k} \|x^{(k+1)} - x^{(k)}\|^2 + \frac{\theta_k}{2t} L^2 \|y^{(k+1)} - y^*\|^2$$

where $L = \|A\|$. Thus, summing up (19) and (18) yields

$$\begin{aligned} &\frac{\|x^{(k+1)} - x^*\|^2}{2\theta_k} + \frac{\|y^{(k+1)} - y^*\|^2}{2\tau_k} \leq \\ &\frac{\|x^{(k)} - x^*\|^2}{2\theta_k} + \frac{\|y^{(k)} - y^*\|^2}{2\tau_k} - (1-t)\frac{\|x^{(k+1)} - x^{(k)}\|^2}{2\theta_k} + \\ &- \frac{\|y^{(k+1)} - y^{(k)}\|^2}{2\tau_k} + \frac{\theta_k}{2t} L^2 \|y^{(k+1)} - y^*\|^2 \end{aligned} \tag{20}$$

In particular, recalling (7), the inequality (20) results in

$$\frac{\|x^{(k+1)} - x^*\|^2}{2\theta_k} \leq \frac{\|x^{(k)} - x^*\|^2}{2\theta_k} + \frac{D^2}{2\tau_k} + \frac{\theta_k}{2t} L^2 D^2$$

By multiplying both sides of the previous inequality by $2\theta_k$ and summing up for $k = 0, ..., N-1$ we obtain

$$\|x^{(N)} - x^*\|^2 \leq \|x^{(0)} - x^*\|^2 + D^2 \sum_{k=0}^{N-1} \frac{\theta_k}{\tau_k} + \frac{L^2 D^2}{t} \sum_{k=0}^{N-1} \theta_k^2$$

Then, the boundedness of $\{x^{(k)}\}$ is ensured by the hypothesis (17). □

Thanks to the previous Lemma, the boundedness of the subgradients $g^{(k)}$ is assured.

**Theorem 3** *Assume that $f_0(x)$ is locally Lipschitz in its domain. If the steplength sequences $\{\theta_k\}$ and $\{\tau_k\}$ satisfy the conditions A1–A3 and* (17)*, then $\{f(x^{(k)})\}$ converges to a minimum of $f(x)$ over $X$ and dist$(x^{(k)}, X^*)$ converges to zero.*

*Proof.* From the previous Lemma and by Proposition 2, the sequence of subgradients $g^{(k)}$ is bounded, thus, from (15) and A1, we have $\|x^{(k)} - x^{(k+1)}\| \to 0$ as $k \to \infty$. If $f_0(x)$ is locally Lipschitz continuous, then for every compact subset $K \subset \mathrm{dom}(f_0^X)$, there exists a positive constant $M_K$ such that $\|f_0(z) - f_0(x)\| \leq M_K \|z - x\|$ for all $x, z \in K$. Thus, since all the iterates are contained in a suitable compact subset of $\mathrm{dom}(f_0^X)$, we have $|f_0^X(x^{(k)}) - f_0^X(x^{(k+1)})| \to 0$ as $k \to \infty$. As consequence, the sequence $\{\delta^k\}$ defined in (16) converges to zero as $k$ diverges. Then, we can invoke Theorem 2 to conclude that $\{f(x^{(k)})\}$ converges to a minimum of $f(x)$ over $X$ and, furthermore, $\lim_{k\to\infty} \mathrm{dist}(x^{(k)}, X^*) = 0$. □

**Remark**. If $X^* = \{x^*\}$, Theorems 2 and 3 state the convergence of the sequences $\{x^{(k)}\}$ generated by the Algorithms 1 and 2 to the unique solution $x^*$ of the problem (5).

In the following section we discuss the implementation of Algorithms 1 and 2 for the TV restoration of images.

## 4 Applications to Total Variation image restoration problems

In this section we consider problem (5) where the objective function is the combination of a convex, proper, lower semicontinuous function measuring the data fidelity with the discrete TV function (1).
In this case we define the function $f_1$ on $\mathbb{R}^{2n}$ such that

$$f_1(y) = \beta \sum_{i=1}^{n} \|y_i\|, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad y_i \in \mathbb{R}^2, i = 1, ..., n$$

where $\beta > 0$ is the regularization parameter. We denote by $A_i \in R^{2\times n}$ the discrete approximation of the gradient of $x$ at the pixel $i$ and by $A \in \mathbb{R}^{2n\times n}$ the following block matrix

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}$$

Then, we have

$$f_1(Ax) = \beta \sum_{i=1}^{n} \|A_i x\| \tag{21}$$

The conjugate of $f_1$ is $f_1^*(y) = \iota_Y(y)$, namely the indicator function of the set $Y \subset \mathbb{R}^{2n}$ defined as follows

$$Y = \left\{ y \in \mathbb{R}^{2n}, y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \ y_i \in \mathbb{R}^2 : \|y_i\| \leq 1, i = 1, ..., n \right\} \tag{22}$$

As a consequence of this, $\mathrm{dom}(f_1^*) = Y$ satisfies (7) and we can write

$$f_1(Ax) = \max_{y \in Y} \beta \langle y, Ax \rangle$$

We remark that the updating rules (10) and (12) for the variable $y$ reduce to the orthogonal projection onto the set $Y$, which is defined by the following closed formula

$$y_i^{(k+1)} = \frac{y_i^{(k)} + \beta \tau_k A_i x^{(k)}}{\max\{1, \|y_i^{(k)} + \beta \tau_k A_i x^{(k)}\|\}}, \quad i = 1, ..., n \tag{23}$$

### 4.1 Gaussian noise

We notice that the PDHG algorithms proposed in [31] are special cases of Algorithm 2 for $f_0(x) = \frac{1}{2}\|Hx-g\|^2$, $X = \mathbb{R}^n$. In this case, denoting by $\mathcal{N}(A)$ and $\mathcal{N}(H)$ the null spaces of $A$ and $H$, under the usual assumption that

$$\mathcal{N}(A)\bigcap\mathcal{N}(H) = \{0\}, \tag{24}$$

the solution of the minimization problem (3) is unique. Thus, Theorem 3 establishes the convergence to this unique solution.
In particular, we stress that the steplength choices indicated by the authors in [31] and employed for the numerical experiments also in [16, 10] guarantee the convergence, according to Theorem 3. Indeed, the proposed steplength choices are

$$\begin{aligned} \tau_k &= 0.2+0.08k \\ \theta_k = \frac{\lambda_k}{1-\lambda_k} &\text{ with } \lambda_k = \left(0.5-\frac{5}{15+k}\right)/\tau_k \end{aligned} \tag{25}$$

for denoising problems, while

$$\theta_k = 0.5/\tau_k \tag{26}$$

for deblurring problems. Both these choices satisfy the hypotheses of Theorem 3, which provides the theoretical foundation to the convergence that, so far, was experimentally observed.
An explicit variant of PDHG can be derived from Algorithm 1, by setting $g^{(k)} = H^T(Hx^{(k)}+b-g)$. For denoising problems, in order to define a bounded set $X$ containing the solution $x^*$, we recall the following result.

**Lemma 3** *Let $x^*$ be the unique minimum point of $f(x) \equiv f_0(x)+f_1(Ax)$, where $f_0(x) = \frac{1}{2}\|x-g\|^2$ and $f_1(Ax)$ is the TV function* (21)*. Then we have*

$$g_{min} \equiv \min_j g_j \le x_i^* \le g_{max} \equiv \max_j g_j$$

*Proof.* See for example [7]. □
We observe that the previous result can be adapted also for constrained problems of the form $\min_{x\ge\eta} f(x)$. In this case, the lower bound of the solution becomes $\max\{\eta, g_{min}\}$, where the maximum is intended componentwise. In summary, if we define

$$X = \{x\in\mathbb{R}^n : \max\{\eta, g_{min}\} \le x \le g_{max}\}$$

Algorithm 1 with $g^{(k)} = x^{(k)} - g$ is convergent to the unique solution $x^*$ of $\min_{x\ge\eta} f(x)$, thanks to Theorem 1.
For general deblurring problems, convergence of Algorithm 1 is ensured under the hypothesis that the generated sequence is bounded.

### 4.2 Poisson noise

Algorithms 1 and 2 can be applied to the denoising or deblurring of images corrupted by Poisson noise, where the data discrepancy is expressed by the generalized KL divergence (4) and $X$ is a subset of the domain of $f_0(x)$. It is well known that $f_0(x)$ is a proper, convex and lower semicontinuous function. Under the hypothesis (24), the solution of the minimization problem (3) on $X$ exists and it is unique. Since $f_0(x)$ is a differentiable function, an explicit version of Algorithm 1 can be implemented by setting $g^{(k)} = \nabla f_0(x^{(k)}) = H^T e - H^T Z(x^{(k)})^{-1} g$, where $e$ is the $n$-vector with all entries equal to one and $Z(x) = \mathrm{diag}(Hx+b)$.
For deblurring problems, in order to state the convergence of Algorithm 1 using Theorem 1, we need to assume that the sequence of the iterates stays bounded.
In case of a denoising problem ($H = I$), the convergence of Algorithm 1 is ensured by defining $X$ as a suitable bounded set containing the unique solution, as suggested in the following Lemma.

**Lemma 4** *Let $x^*$ be the unique solution of the problem*

$$\min_{x\ge 0} f(x) \equiv f_0(x)+f_1(Ax), \tag{27}$$

*where $f_0(x) = \sum_i g_i \log\frac{g_i}{x_i} + x_i - g_i$, with $g_i\log g_i = 0$ if $g_i = 0$, and $f_1(Ax)$ is the TV function* (21)*. Then, for all $i$ such that $g_i > 0$ we have*

$$g_{min} \equiv \min\{g_j : g_j > 0\} \le x_i^* \le g_{max} \equiv \max_j g_j$$

*Proof.* See [6]. □
Consequently, for denoising problems from data corrupted by Poisson noise, if we define

$$\begin{aligned} X = \{x\in\mathbb{R}^n : & g_{min} \le x_i \le g_{max} \text{ for } g_i > 0, \\ & 0 \le x_i \le g_{max} \text{ otherwise}\} \end{aligned}$$

then, the unique solution $x^*$ of (27) belongs to $X$ and the sequence generated by Algorithm 1 converges to $x^*$.
Furthermore, it is easy to see that the $i$-th component of $(I+\theta\partial f_0)^{-1}(p)$ is given by

$$(I+\theta\partial f_0)^{-1}(p)_i = \frac{1}{2}\left(p_i-\theta+\sqrt{(\theta-p_i)^2+4\theta g_i}\right)$$

which can be exploited for the computation of the step (13) in Algorithm 2.
For the deblurring problems, a closed form formula for (13) is not available, thus Algorithm 2 can be difficult to implement in this case.

### 4.3 Impulse noise

When the noise affecting the data contains strong outliers (e.g. impulse or salt&pepper noise), a well suited data discrepancy function is the non–smooth $L^1$ norm:

$$f_0(x) = \|x-g\|_1 = \sum_{k=1}^{n} |x_i - g_i| \tag{28}$$

The resolvent operator of $f_0$ is given by the pointwise shrinkage operations:

$$(I+\theta\partial f_0)^{-1}(p)_i = \begin{cases} p_i-\theta & \text{if } p_i - g_i > \theta \\ p_i+\theta & \text{if } p_i - g_i < -\theta \\ g_i & \text{if } |p_i-g_i| \le \theta \end{cases} \tag{29}$$

This closed-form representation of the resolvent operator $(I+\theta\partial f_0)^{-1}$ enables us to apply Algorithm 2 to the minimization of the $L^1$–TV model (28)–(21).

### 4.4 Smoothed Total Variation

In many papers (see, for example, [11], [2], [3], [30]), the discrete TV (21) is replaced by the following smoothed approximation

$$f_1(Ax) = \beta \sum_{i=1}^{n} \left\| \begin{pmatrix} A_i x \\ \rho \end{pmatrix} \right\| \tag{30}$$

where $\rho$ is a small positive parameter. This variant has been considered also in a general edge–preserving regularization framework as Hypersurface (HS) potential [12]. When the smoothed TV regularization (30) is used, if $f_0(x)$ is a differentiable function, the minimization of $f(x)$ can be obtained by efficient differentiable optimization methods. In this section we show that Algorithms 1 and 2 can also be applied with minor modifications.
In this case the conjugate is the indicator function of the following set

$$Y = \left\{ y \in \mathbb{R}^{3n}, y = \begin{pmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{pmatrix}, \ \tilde{y}_i \in \mathbb{R}^3 : \|\tilde{y}_i\| \le 1, i = 1,...,n \right\}$$

In particular, setting $\tilde{y}_i = \begin{pmatrix} y_i \\ z_i \end{pmatrix}$, with $y_i \in \mathbb{R}^2$, Algorithms 1 and 2 can be adapted to the smoothed TV model by modifying the projection at the steps (10) and (12), detailed in (23), as follows:

$$y_i^{(k+1)} = \frac{y_i^{(k)} + \beta\tau_k A_i x^{(k)}}{d_i^{(k)}}, \ \ z_i^{(k+1)} = \frac{z_i^{(k)} + \beta\tau_k\rho}{d_i^{(k)}}$$

for $i = 1,...,n$, where

$$d_i^{(k)} = \max\left\{ 1, \left\| \begin{pmatrix} y_i^{(k)} + \beta\tau_k A_i x^{(k)} \\ z_i^{(k)} + \beta\tau_k\rho \end{pmatrix} \right\| \right\}$$

As observed in [6], numerical methods which are not based on the differentiability of (30) can be convenient for the smoothed TV minimization, especially when the smoothing parameter $\rho$ is very small.

## 5 Numerical Experience

The steplength selection is a crucial issue for the practical performance of many algorithms. In particular, as pointed out in [31], some choices yield a fast initial convergence, but they are less suited to achieve fast asymptotic convergence and viceversa. One of the main strength of the proposed method is that variable steplength parameters are allowed, and this can help to achieve both fast initial and asymptotic convergence. In the case of the quadratic data fidelity function (2), Algorithm 2 with the choices (25)-(26) is equivalent to the PDHG method and its performances has been experimentally evaluated in [31,16,10], showing that convenient steplength sequences lead to a very appealing efficiency with respect to the state–of–the–art methods.
This section is devoted to numerically evaluate the effectiveness of Algorithm 1 for TV restoration of images corrupted by Poisson noise. Algorithm 1 and 2 can be used also for other imaging problems. At the end of the section, we show as Algorithm 2 can be applied for solving an image denoising problem with impulse noise.
The numerical experiments described in this section have been performed in MATLAB environment, on a server with a dual Intel Xeon QuadCore E5620 processor at 2,40 GHz, 12 Mb cache and 18 Gb of RAM.

### 5.1 Poisson noise

We compare Algorithm 1 with two methods, especially tailored for the TV restoration in presence of data corrupted by Poisson noise. The first one is the PIDSplit+ algorithm [26], based on a very efficient alternating split Bregman technique. The algorithm guarantees that the approximate solution satisfies the constraints and it depends only on a positive parameter $\gamma$. The second one is the Alternating Extragradient Method (AEM) [6], that solves the primal–dual formulation of the problem (3) by a successive updating of dual and primal iterates in an alternating, or Gauss-Seidel, way. Algorithm 1 and AEM are very similar, but AEM requires an additional ascent (extragradient) step at any iteration and it employs the same adaptively computed steplength in the three steps. Furthermore, in the denoising experiments, we include in the comparison also the algorithm in [10, Algorithm 1, p. 122], with $\theta = 1$, that in the following is denoted by CP. This general primal–dual algorithm requires that the resolvent operators of $f_0$ and $f_1^*$ have a closed–form representation. As observed in subsection 4.2, the resolvent of the

Kullback–Leibler divergence is easy to compute only for the denoising case. The CP method depends on two parameters $\sigma, \tau$, which should be chosen such that $\sigma\tau L^2 = 1$, where $\beta\|A\| \leq L$.

In the experiments we consider a set of test problems, where the Poisson noise has been simulated by the `imnoise` function in the Matlab Image Processing Toolbox. The considered test problems are described in the following.

*Denoising test problems*

- LCR phantom: the original image is the phantom described in [21]; it is an array $256 \times 256$, consisting in concentric circles of intensities 70, 135 and 200, enclosed by a square frame of intensity 10, all on a background of intensity 5 (LCR-1). We can simulate a different noise level by multiplying the LCR phantom by a factor 10 (LCR-10) and 0.2 (LCR-0.2) and generating the corresponding noisy images. The relative difference in $l_2$ norm between the noisy and the original images is 0.095318, 0.030266, 0.21273 respectively.
- Airplane (AIR): the original image is an array $256 \times 256$ (downloadable from *http://sipi.usc.edu/database/*), with values in the range $[0, 232]$; the relative difference in $l_2$ norm between the noisy and noise–free images is 0.070668.
- Dental Radiography (DR): the original image [30] is an array $512 \times 512$, with values in the range $[0, 255]$; for this test problem, simulating a radiographic image obtained by a lower dose, the relative difference in $l_2$ norm between the noisy and the noise–free images is 0.17866.

*Deblurring test problems*

- *micro*: the original image is the confocal microscopy phantom of size $128 \times 128$ described in [28]; its values are in the range $[0, 70]$ and the total flux is 2.9461 $10^5$; the background term $b$ in (4) is set to zero.
- *cameraman*: following [26], the simulated data are obtained by convolving the image $256 \times 256$ with a Gaussian psf with standard deviation $\sigma = 1.3$, then adding Poisson noise; the values of the original image are in the range $[0, 1000]$; the background term $b$ in (4) is set to zero.

In the first set of experiments we compare the numerical behavior of Algorithm 1, AEM, PIDSplit+ and CP on the denoising test problems described above. In order to compare the convergence rate from the optimization point of view, we compute the ideal solution $x^*$ of the minimization problem, by running 100000 iterations of AEM. Then, we evaluate the progress toward the ideal solution at each iteration in terms of the $l_2$ relative error

$$e^k = \frac{\|x^{(k)} - x^*\|}{\|x^*\|}$$

It is noticed that computing the ideal solution with AEM makes a small bias in favour of AEM itself. However the obtained results are sufficiently convincing to forget this bias.

In Table 1 for any test problem we report the number of iterations *it* and, in brackets, the computational *time* in seconds needed to satisfy the inequality

$$e^k \leq \mu, \tag{31}$$

for different values of the tolerance $\mu$. We report also the $l_2$ relative reconstruction error

$$e^{rec} = \frac{\|x^\mu - \tilde{x}\|}{\|\tilde{x}\|}$$

where $\tilde{x}$ is the original image and $x^\mu$ is the reconstruction corresponding to the tolerance $\mu$. The symbol * denotes that 3000 iterations have been performed without satisfying the inequality (31). All methods have been initialized with $x^{(0)} = \max\{\eta, g\}$, where the maximum is intended componentwise and $\eta_i = 0$ for $g_i = 0$, $\eta_i = g_{min}$ otherwise, and the constraint set $X$ is defined as in section 4.2. In Algorithm 1, AEM and CP, the initial guess for the dual variable $y^{(0)}$ has been set to zero. For the initial setting of the others variables involved in PIDSplit+ see [26]; the value of $\gamma$ used for the different runs of PIDSplit+ is detailed in Table 1. The value of $\gamma$ suggested by the authors in [26] is $\frac{50}{\beta}$. In Table 1 we report also the value of $\tau$ used for the different runs of CP. In Table 2 we report the sequences chosen for the steplength parameters in Algorithm 1, which have been optimized for the different test problems.

Figure 1 shows the convergence speed of the considered methods for each test problem: we plot, in log-scale, the $l_2$ relative error $e^k$ with respect to the computational time. In Tables 3 and 4 we report some results obtained by running each method for 3000 iterations. In particular, Table 3 shows the $l_2$ norm of the relative error and corresponding computational time after 3000 iterations. We report also, for any test-problem, the relative reconstruction error and the value of the objective function $f(x^{(3000)})$. In order to show the quality of the restored images, in Figure 2 we report the superposition of the row 128 of the original image LCR-0.2 and the related noisy image; besides, we show the same row of the reconstruction corresponding to $\mu = 10^{-4}$, obtained by Algorithm 1.

In Figures 3 and 4 we report the original, noisy and reconstructed images related to test problems AIR and DR. Since at the tolerance $\mu = 10^{-4}$ all the restored images are visually the same, we report only the one provided by the two fastest algorithms. The results of this numerical experimentation on denoising problems suggests the following considerations.

- For denoising problems, Algorithm 1 performs well with respect to the other methods; indeed for careful choices of the steplength parameters, the method exhibits a fast initial convergence and this good behavior is preserved

also asymptotically; furthermore a single iteration is faster than in the other methods.
- About PIDSplit+, depending on the value of the parameter $\gamma$, we can observe either an initial or an asymptotic fast convergence, but we unlikely obtain both the behaviors. Furthermore, any iteration involves the solution of a linear system. Although this operation is performed by fast transforms, the computational complexity of each iteration of PIDSplit+ is greater than in the other methods, where only matrix–vector products and simple projections are performed.
- The choice of the steplength sequences is crucial for the effectiveness of Algorithm 1, as well as the choice of $\gamma$ for PIDSplit+ and of $\tau$ for CP. The numerical experience shows that Algorithm 1 is more sensible to the choice of the sequence $\{\theta_k\}$ rather than $\{\tau_k\}$; this can be explained by observing that the projection over the domain $Y$, due to its special structure (22), may overcome the effect of a too large steplength. Furthermore, about the choice of $\theta_k$, if we set $\theta_k = \frac{1}{ak+b}$, we observe that $b$ affects the initial rate of convergence while the asymptotic behavior of the method is governed by the value of $a$.

The second set of experiments concerns the deblurring test problems *micro* and *cameraman*. We compare Algorithm 1, AEM and PDSplit+, considering as ideal solution $x^*$ the image obtained by running 100000 iterations of AEM. We consider the same initial setting used for the denoising test problems, except for the primal variable: since for deblurring the constraints are $x \geq 0$, then we set $x^{(0)} = \max\{0, g\}$. In Table 4 for each method we report the computational time and the relative error $e^{3000}$ with respect to the ideal solution after 3000 iterations. We report also the minimum value $f(x^{(3000)})$ and the $l_2$ norm of relative reconstruction error $e^{rec}$ (since all methods produce the same optimal values, we report them only once). In Figure 5 we compare the convergence rate of the considered methods for each test problem, by plotting in log-scale the $l_2$ relative error $e^k$ with respect to the computational time. In Figures 6 and 7 we show the restored images obtained after 3000 iterations by Algorithm 1 and PidSplit+ with the more effective choice of $\gamma$.
For deblurring problems, the PIDSplit+ method exhibits a very fast convergence. Indeed, the operator $H$ could be severely ill conditioned, and in this case the implicit step in PIDSplit+ may help to improve the convergence rate. We observe also that for deblurring problems, the complexity of a single iteration of the three methods is similar, since the matrix–vector products involving the matrix $H$ require the same fast transforms employed to compute the implicit step in PIDSplit+.

**Table 2** Steplength sequences chosen for Algorithm 1.

| test problem | $\tau_k$ | $\theta_k$ |
|---|---|---|
| LCR-1 | $0.4+0.01k$ | $\frac{1}{0.0015k+0.15}$ |
| LCR-10 | $0.4+0.01k$ | $\frac{1}{10^{-4}k+0.01}$ |
| LCR-0.2 | $0.9+0.009k$ | $\frac{1}{0.009k+0.7344}$ |
| AIR | $0.9+0.01k$ | $\frac{1}{5\ 10^{-5}k+0.01}$ |
| DR | $0.5+0.01k$ | $\frac{1}{0.0015k+0.15}$ |
| *micro* | $0.9+0.01k$ | $\frac{1}{2\ 10^{-4}k+0.33}$ |
| *cameraman* | $0.9+0.01k$ | $\frac{1}{10^{-5}k+0.04}$ |

### 5.2 Impulse noise

In this section we apply Algorithm 2 to the $L^1$–TV problem described in Section 4.3. The $512 \times 512$ original image *b*oat (downloadable from *http://sipi.usc.edu/database/*) has been corrupted by 25% salt&pepper noise (after adding noise, the image was rescaled so that its value are between 0 and 1). The original and the noisy images are shown in Figure 8, together with the restoration provided by Algorithm 2 after 2000 iterations. For this test problem we set $\beta = 0.65$ and the sequences for the steplength parameters are $\tau_k = 0.1+0.1k$ and $\theta_k = \frac{1}{0.05k+0.1}$. Table 5 shows the results of a comparison between Algorithm 2 and CP ($\theta = 1, \tau = 0.02$). Since the solution of the minimization problem (28)–(21) is in general not unique, we evaluate the normalized error on the primal objective function $E^k = \frac{f(x^{(k)})-f(x^*)}{f(x^*)}$, where $x^*$ is computed by running 100000 iterations of CP. In Table 5 we report the number of iterations *it* and the CPU time in seconds needed to drop the normalized error $E^k$ below the tolerance $\mu$. Figure 9 shows the convergence speed of Algorithm 2 and CP for the test problem *b*oat: we plot in log–scale the normalized error $E^k$ with respect to the computational time.

**Table 5** Image denoising problems in the case of impulse noise: test problem *boat*, $\beta = 0.65$; comparison between Algorithm 2 and CP.

| | $\mu = 10^{-4}$ | $\mu = 10^{-5}$ |
|---|---|---|
| method | *it (time)* | *it (time)* |
| Algorithm 2 | 370(38.87) | 884(93.98) |
| CP | 186(23.8) | 340(43.60) |

## 6 Conclusions

The main contribution of this paper is the analysis of a class primal–dual algorithms for convex optimization problems, providing the related convergence proof based on the $\varepsilon$-subgradient techniques. The developed analysis applies, as a special case, to the PDHG method [31], but allows further generalizations. Indeed, as immediate consequence, new meth-

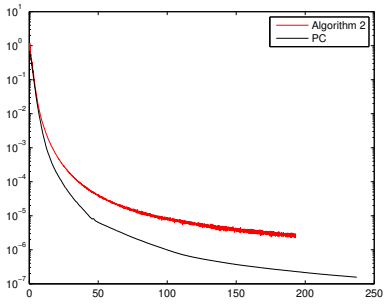



**Fig. 9** Image denoising in the case of impulse noise: test problem *boat*. Convergence speed of Algorithm 2 and PC ($\tau = 0.02$): plot of normalized error $E^k$ versus the computational time. All the methods run for 2000 iterations.

ods for solving TV denoising and deblurring problems can be derived from the basic scheme. These methods can be applied also to other imaging problems based on convex regularization terms and data discrepancy functions with simple resolvent operator (see [10] for some examples of applications).

The crucial point to obtain fast convergence is the selection of *a priori* steplength sequences: numerical experiments show that there exists clever choices leading to very appealing performances (at the initial iterations and also asymptotically), comparable to other state-of-the-art methods.

The numerical experimentation highlights also the importance of the parameters choice for the convergence behaviour of all the considered methods.

For image denoising problems, where the relationship between image and object is simpler, the proposed scheme seems very efficient in terms of accuracy and computational complexity, although the choice of convenient steplength sequences is a difficult problem.

Future work will concern the design of an adaptive rule for generating the steplength sequences, following a strategy similar to that suggested in [5] for the subgradient method.

**Acknowledgements** We are grateful to the anonymous referees for their comments, which stimulated us to greatly improve the paper.

## References

1. Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in Linear and Non-Linear Programming, vol. II. Stanford University Press, Stanford (1958)
2. Aujol, J.F.: Some first-order algorithms for total variation based image restoration. J. Math. Imaging Vis. **34**(3), 307–327 (2009)
3. Bardsley, J.M., Luttman, A.: Total variation–penalized Poisson likelihood estimation for ill–posed problems. Adv. Comput. Math. **31**(1–3), 35–59 (2009)
4. Beck, A., Teboulle, M.: Fast gradient–based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Processing **18**, 2419–2434 (2009)
5. Bertsekas, D.: Convex Optimization Theory. Athena Scientific (2009)
6. Bonettini, S., Ruggiero, V.: An alternating extragradient method for total variation based image restoration from Poisson data. Inverse Problems **27**, 095,001 (2011)
7. Brune, C., Sawatzky, A., Wübbeling, F., Kösters, T., Burger, M.: Forward-Backward EM-TV methods for inverse problems with Poisson noise. http://wwwmath.uni-muenster.de/num/publications/2009/BBSKW09/ (2009)
8. Chambolle, A.: An algorithm for Total Variation minimization and applications. J. Math. Imag. Vis. **20**, 89–97 (2004)
9. Chambolle, A.: Total variation minimization and a class of binary MRF models. EMMCVPR 05. Lecture Notes in Computer Sciences **3757**, 136–152 (2005)
10. Chambolle, A., Pock, T.: A first–order primal–dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**, 120–145 (2011)
11. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal–dual method for Total Variation based image restoration. SIAM J. Sci. Comput. **20**, 1964–1977 (1999)
12. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, A.: Deterministic edge–preserving regularization in computed imaging. IEEE Trans. Image Processing **6** (1997)
13. Correa, R., Lemaréchal, C.: Convergence of some algorithms for convex minimization. Math. Prog. **62**, 261–275 (1993)
14. Dahl, J., Hansen, P.C., Jensen, S.H., Jensen, T.L.: Algorithms and software for total variation image reconstruction via first-order methods. Numerical Algorithms **53**, 67–92 (2010)
15. Ekeland, I., Témam, R.: Convex Analysis and Variational Problems. SIAM (1999)
16. Esser, E., Zhang, X., Chan, T.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. SIAM J. Imaging Sci. **3**(4), 1015–1046 (2010)
17. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. IEEE Trans. Pattern Analysis and Machine Intelligence **6**, 721–741 (1984)
18. Goldfarb, D., Yin, W.: Second–order cone programming methods for total variation based image restoration. SIAM J. Sci. Comput. **27(2)**, 622–645 (2005)
19. Goldstein, T., Osher, S.: The split Bregman method for L1 regularized problems. SIAM Journal on Imaging Sciences
20. Larsson, T., Patriksson, M., Strömberg, A.B.: On the convergence of conditional $\varepsilon$-subgradient methods for convex programs and convex–concave saddle-point problems. European Journal of Operational Research **151**, 461–473 (2003)
21. Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. J. Math. Imaging Vis. **27**, 257–263 (2007)
22. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. SIAM Journal on Multiscale Modeling and Simulation **4**(2), 460–489 (2005)
23. Robinson, S.M.: Linear convergence of epsilon-subgradient descent methods for a class of convex functions. Math. Program., Ser. A **86** (1999)
24. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton, NJ (1970)
25. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
26. Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. J. Vis. Commun. Image R. **21**, 193–199 (2010)
27. Shepp, L.A., Vardi, Y.: Maximum likelihood reconstruction for emission tomography. Trans. Med. Imaging **MI–1**, 113–122 (1982)
28. Willett, R.M., Nowak, R.D.: Platelets: A multiscale approach for recovering edges and surfaces in photon limited medical imaging. IEEE Transactions on Medical Imaging **22**, 332–350 (2003)

29. Yu, G., Qi, L., Dai, Y.: On nonmonotone Chambolle gradient projection algorithms for total variation image restoration. J. Math. Imaging Vis. **35** (2009)
30. Zanella, R., Boccacci, P., Zanni, L., Bertero, M.: Efficient gradient projection methods for edge-preserving removal of Poisson noise. Inverse Problems **25** (2009)
31. Zhu, M., Chan, T.F.: An efficient primal–dual hybrid gradient algorithm for Total Variation image restoration. CAM Report 08-34, UCLA (2008)
32. Zhu, M., Wright, S.J., Chan, T.F.: Duality-based algorithms for total-variation-regularized image restoration. Computational Optimization and Applications **47**, 377–400 (2008)

**Table 1** Denoising problems: comparison between Algorithm 1, AEM and PIDSplit+.

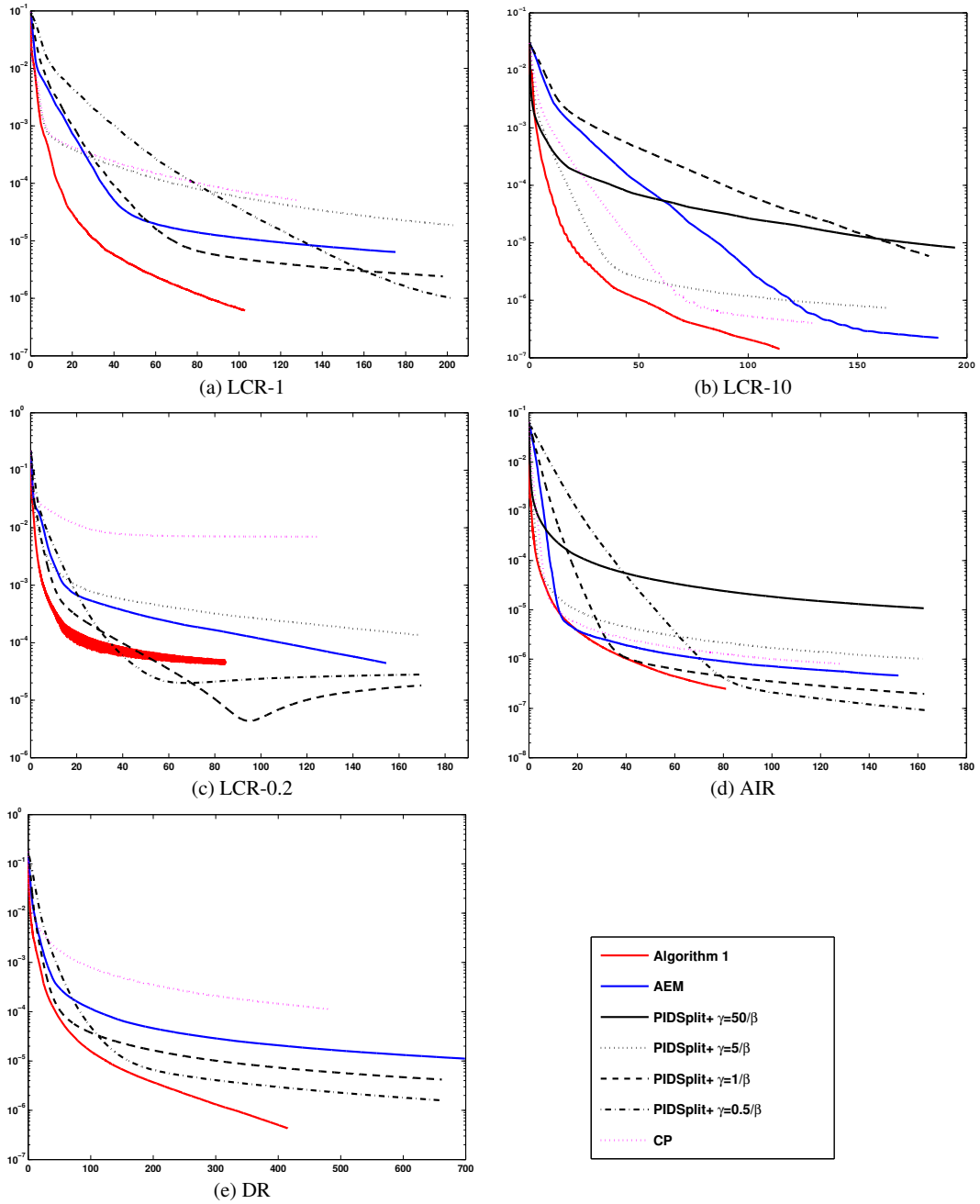
| method | *it (time)* | $e^{rec}$ | *it (time)* | $e^{rec}$ | *it (time)* | $e^{rec}$ | *it (time)* | $e^{rec}$ |
|---|---|---|---|---|---|---|---|---|
| | LCR-1, $\beta = 0.25$ | | | | | | | |
| | $\mu = 10^{-2}$ | | $\mu = 10^{-3}$ | | $\mu = 10^{-4}$ | | $\mu = 5\ 10^{-6}$ | |
| Algorithm 1 | 43 (1.8) | 0.02635 | 137 (5.2) | 0.02488 | 377 (13.2) | 0.02497 | 1239 (43.0) | 0.02498 |
| AEM | 63 (3.3) | 0.02306 | 312 (17.5) | 0.02468 | 604 (32.8) | 0.02495 | * | * |
| PIDSplit+ $\gamma = \frac{5}{\beta}$ | 26 (1.5) | 0.02505 | 99 (6.1) | 0.02502 | 1022(60.7) | 0.02499 | * | * |
| PIDSplit+ $\gamma = \frac{1}{\beta}$ | 81 (4.9) | 0.02282 | 301 (17.9) | 0.02460 | 593 (35.6) | 0.02496 | 1477 (88.7) | 0.02498 |
| PIDSplit+ $\gamma = \frac{0.5}{\beta}$ | 157 (9.4) | 0.02270 | 601 (35.7) | 0.02459 | 1182 (70.5) | 0.02496 | 2185 (130.6) | 0.02498 |
| CP $\tau = 5$ | 37 (1.5) | 0.026031 | 173 (7.2) | 0.025021 | 1917(80.7) | 0.024985 | * | * |
| CP $\tau = 2$ | 43 (1.7) | 0.023419 | 246(10.5) | 0.024722 | 654(27.9) | 0.024981 | * | * |
| CP $\tau = 0.8$ | 90 (3.7) | 0.022565 | 570(24.8) | 0.024643 | 1118(48.2) | 0.024947 | 2099(89.8) | 0.024983 |
| | LCR-10, $\beta = 0.05$ | | | | | | | |
| | $\mu = 10^{-2}$ | | $\mu = 10^{-3}$ | | $\mu = 10^{-4}$ | | $\mu = 10^{-6}$ | |
| Algorithm 1 | 6 (0.2) | 0.01219 | 69 (2.6) | 0.008464 | 194 (6.9) | 0.008386 | 1254 (49.3) | 0.008383 |
| AEM | 91 (5.0) | 0.01188 | 326 (17.4) | 0.008073 | 815 (42.2) | 0.008357 | 1942 (98.1) | 0.008383 |
| PIDSplit+ $\gamma = \frac{50}{\beta}$ | 4 (0.3) | 0.01106 | 61 (3.4) | 0.008502 | 539 (30.6) | 0.008386 | * | * |
| PIDSplit+ $\gamma = \frac{5}{\beta}$ | 23 (1.3) | 0.01199 | 113 (6.0) | 0.007928 | 319 (17.4) | 0.008338 | 2150 (117.8) | 0.008383 |
| PIDSplit+ $\gamma = \frac{1}{\beta}$ | 115 (6.5) | 0.01192 | 556 (30.6) | 0.007915 | 1591 (89.2) | 0.008338 | * | * |
| CP $\tau = 20$ | 21(0.8) | 0.012146 | 123(5.2) | 0.008129 | 327(14.0) | 0.008361 | 2948(128.7) | 0.008383 |
| CP $\tau = 10$ | 41(1.7) | 0.012084 | 229(9.3) | 0.008078 | 629(26.1) | 0.008357 | 1646(68.9) | 0.008383 |
| CP $\tau = 1$ | 408(17.8) | 0.011906 | 2238(96.6) | 0.008062 | * | * | * | * |
| | LCR-0.2, $\beta = 0.575$ | | | | | | | |
| | $\mu = 10^{-2}$ | | $\mu = 10^{-3}$ | | $\mu = 10^{-4}$ | | $\mu = 5\ 10^{-5}$ | |
| Algorithm 1 | 58 (1.7) | 0.04442 | 189 (5.4) | 0.04453 | 706 (20.2) | 0.04476 | 1881 (54.0) | 0.04477 |
| AEM | 101 (5.5) | 0.04413 | 279 (14.8) | 0.04466 | 2103 (108.8) | 0.04477 | 2870 (147.9 | 0.04477 |
| PIDSplit+ $\gamma = \frac{5}{\beta}$ | 59 (3.1) | 0.04660 | 355 (20.0) | 0.04483 | * | * | * | * |
| PIDSplit+ $\gamma = \frac{1}{\beta}$ | 67 (3.8) | 0.04389 | 174 (10.0) | 0.04466 | 695 (39.6) | 0.04477 | 931 (53.1) | 0.04477 |
| PIDSplit+ $\gamma = \frac{0.5}{\beta}$ | 118 (6.7) | 0.04322 | 317 (18.0) | 0.04458 | 605 (34.2) | 0.04477 | 743 (42.2) | 0.04477 |
| CP $\tau = 10$ | 577(24.2) | 0.046580 | * | * | * | * | * | * |
| CP $\tau = 1$ | 102( 4.2) | 0.045886 | * | * | * | * | * | * |
| CP $\tau = 0.1$ | 461(19.2) | 0.043074 | * | * | * | * | * | * |
| | AIR, $\beta = 0.05$ | | | | | | | |
| | $\mu = 10^{-2}$ | | $\mu = 10^{-3}$ | | $\mu = 10^{-4}$ | | $\mu = 10^{-6}$ | |
| Algorithm 1 | 4 (0.1) | 0.02440 | 36 (1.0) | 0.02138 | 122 (3.4) | 0.02137 | 1521 (40.7) | 0.02137 |
| AEM | 67 (3.2) | 0.02589 | 133 (6.0) | 0.02168 | 191 (8.6) | 0.02140 | 1448 (72.4) | 0.02137 |
| PIDSplit+ $\gamma = \frac{50}{\beta}$ | 6 (0.3) | 0.02271 | 60 (3.1) | 0.02141 | 450 (24.2) | 0.02137 | * | * |
| PIDSplit+ $\gamma = \frac{5}{\beta}$ | 17 (0.9) | 0.02529 | 39 (2.0) | 0.02158 | 74 (3.9) | 0.02138 | * | * |
| PIDSplit+ $\gamma = \frac{0.5}{\beta}$ | 162 (8.6) | 0.02587 | 378 (20.5) | 0.02163 | 645 (35.1) | 0.02139 | 1299 (70.7) | 0.02137 |
| CP $\tau = 20$ | 16(0.7) | 0.026173 | 41(1.8) | 0.021622 | 96 (4.4) | 0.021372 | * | * |
| CP $\tau = 10$ | 30(1.4) | 0.026222 | 76(3.5) | 0.021685 | 127 (5.7) | 0.021400 | 2361(103.1) | 0.021374 |
| CP $\tau = 1$ | 288(12.2) | 0.026105 | 732(31.8) | 0.021698 | 1223(53.7) | 0.021403 | 2268(99.4) | 0.021374 |
| | DR, $\beta = 0.27$ | | | | | | | |
| | $\mu = 10^{-2}$ | | $\mu = 10^{-3}$ | | $\mu = 10^{-4}$ | | $\mu = 5\ 10^{-6}$ | |
| Algorithm 1 | 25 (3.4) | 0.02900 | 121 (17.5) | 0.02961 | 312 (43.8) | 0.02965 | 1258 (173.7) | 0.02966 |
| AEM | 42 (9.7) | 0.02753 | 121 (29.3) | 0.02949 | 480 (111.2) | 0.02966 | * | * |
| PIDSplit+ $\gamma = \frac{5}{\beta}$ | 18 (3.8) | 0.02813 | 141 (30.2) | 0.02956 | 920 (199.3) | 0.02965 | * | * |
| PIDSplit+ $\gamma = \frac{1}{\beta}$ | 46 (10.1) | 0.02880 | 106 (24.0) | 0.02952 | 233 (52.4) | 0.02966 | 2566 (566.7) | 0.02966 |
| PIDSplit+ $\gamma = \frac{0.5}{\beta}$ | 89 (19.6) | 0.02915 | 202 (45.0) | 0.02953 | 377 (83.1) | 0.02966 | 1130 (249.8) | 0.02966 |
| PC $\tau = 10$ | 45( 7.0) | 0.027906 | 518(81.7) | 0.029568 | * | * | * | * |
| CP $\tau = 1$ | 45( 6.8) | 0.027506 | 148(23.2) | 0.029435 | 402(63.0) | 0.029655 | * | * |
| CP $\tau = 0.1$ | 386(62.4) | 0.028091 | 1294(209.3) | 0.029385 | 2453(399.7) | 0.029639 | * | * |

**Table 3** Denoising problems: $l_2$ norm of the relative error and computational time after 3000 iterations.

| | optimal values | | Algorithm 1 | | AEM | | PIDSplit+ | | | | | | CP | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $e^{rec}$ | $f(x^{(3000)})$ | $e^{3000}$ | *time* | $e^{3000}$ | *time* | $\gamma$ | $e^{3000}$ | *time* | $\gamma$ | $e^{3000}$ | *time* | $\tau$ | $e^{3000}$ | *time* |
| LCR-1 $\beta = 0.25$ | 0.02498 | 54848.8 | $6.1\ 10^{-7}$ | 102.8 | $6.4\ 10^{-6}$ | 170.3 | $\frac{1}{\beta}$ | $2.4\ 10^{-6}$ | 179.9 | $\frac{0.5}{\beta}$ | $1.2\ 10^{-6}$ | 178.5 | 0.8 | $3.3\ 10^{-6}$ | 127.1 |
| LCR-10 $\beta = 0.05$ | 0.008383 | 76457.2 | $1.4\ 10^{-7}$ | 114.3 | $2.2\ 10^{-7}$ | 151.0 | $\frac{5}{\beta}$ | $7.4\ 10^{-7}$ | 164.1 | $\frac{1}{\beta}$ | $5.9\ 10^{-6}$ | 182.6 | 10 | $4.1\ 10^{-7}$ | 126 |
| LCR-0.2 $\beta = 0.575$ | 0.004477 | 44434.1 | $4.6\ 10^{-5}$ | 84.5 | $4.4\ 10^{-5}$ | 154.4 | $\frac{1}{\beta}$ | $1.8\ 10^{-5}$ | 169.6 | $\frac{0.5}{\beta}$ | $2.8\ 10^{-5}$ | 169.9 | 10 | $7.2\ 10^{-3}$ | 124.6 |
| AIR $\beta = 0.05$ | 0.02137 | 43482.3 | $2.5\ 10^{-7}$ | 81.1 | $4.7\ 10^{-7}$ | 151.8 | $\frac{1}{\beta}$ | $2.0\ 10^{-7}$ | 162.6 | $\frac{0.5}{\beta}$ | $9.1\ 10^{-8}$ | 164.2 | 1 | $8.6\ 10^{-8}$ | 131.3 |
| DR $\beta = 0.27$ | 0.02966 | 146803 | $4.3\ 10^{-7}$ | 415.4 | $1.1\ 10^{-5}$ | 699.1 | $\frac{1}{\beta}$ | $4.2\ 10^{-6}$ | 661.7 | $\frac{0.5}{\beta}$ | $1.6\ 10^{-6}$ | 663.6 | 1 | $9.2\ 10^{-6}$ | 481.1 |

**Table 4** Deblurring problems: $l_2$ norm of the relative error and computational time after 3000 iterations of the considered methods.

| test | method | *time* | $e^{3000}$ | $e^{rec}$ | $f(x^{(3000)})$ |
|---|---|---|---|---|---|
| `micro` | Algorithm 1 | 51.1 | 0.01207 | 0.09138 | 11112.4 |
| | AEM | 74.7 | 0.01161 | 0.09156 | 11112.0 |
| | PIDSplit+ $\gamma = 50/\beta$ | 85.5 | 0.0008703 | 0.09227 | 11113.4 |
| | PIDSplit+ $\gamma = 5/\beta$ | 89.2 | 0.001661 | 0.09227 | 11111.9 |
| | PIDSplit+ $\gamma = 1/\beta$ | 90.8 | 0.008793 | 0.09212 | 11111.9 |
| *cameraman* | Algorithm 1 | 177.7 | $1.023\ 10^{-3}$ | 0.08744 | 9949.2 |
| | AEM | 240.1 | $1.046\ 10^{-3}$ | 0.08744 | 9948.8 |
| | PIDSplit+ $\gamma = 50/\beta$ | 310.2 | $1.308\ 10^{-4}$ | 0.08751 | 9948.9 |
| | PIDSplit+ $\gamma = 5/\beta$ | 306.3 | $9.733\ 10^{-7}$ | 0.08751 | 9948.8 |
| | PIDSplit+ $\gamma = 1/\beta$ | 306.0 | $1.793\ 10^{-4}$ | 0.08751 | 9948.8 |




(a) LCR-1 (b) LCR-10

(c) LCR-0.2 (d) AIR

(e) DR

**Fig. 1** Denoising problems: convergence speed of Algorithm 1, AEM, PIDSplit+ (with different values of $\gamma$) and CP (with $\tau = 10$ in all test problems except (a) where $\tau = 5$): plot of $l_2$ norm of the relative error $e^k$ with respect to the ideal solution of the minimization problem versus the computational time. All the methods run for 3000 iterations.

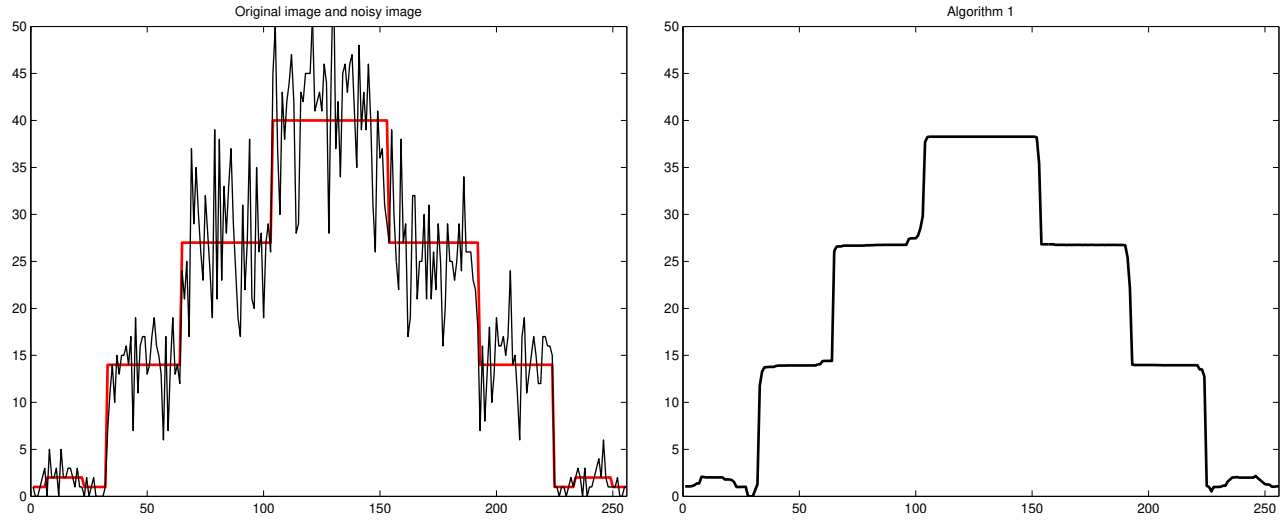
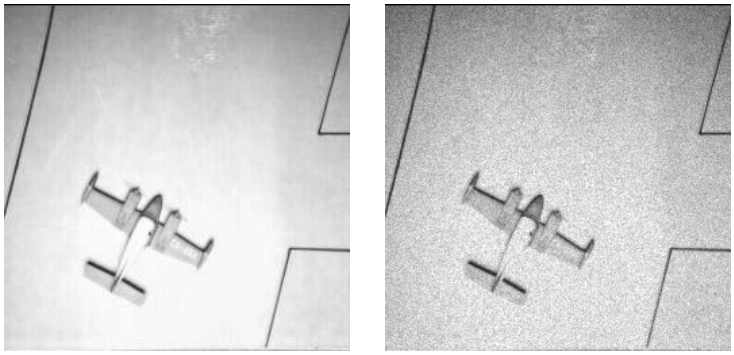


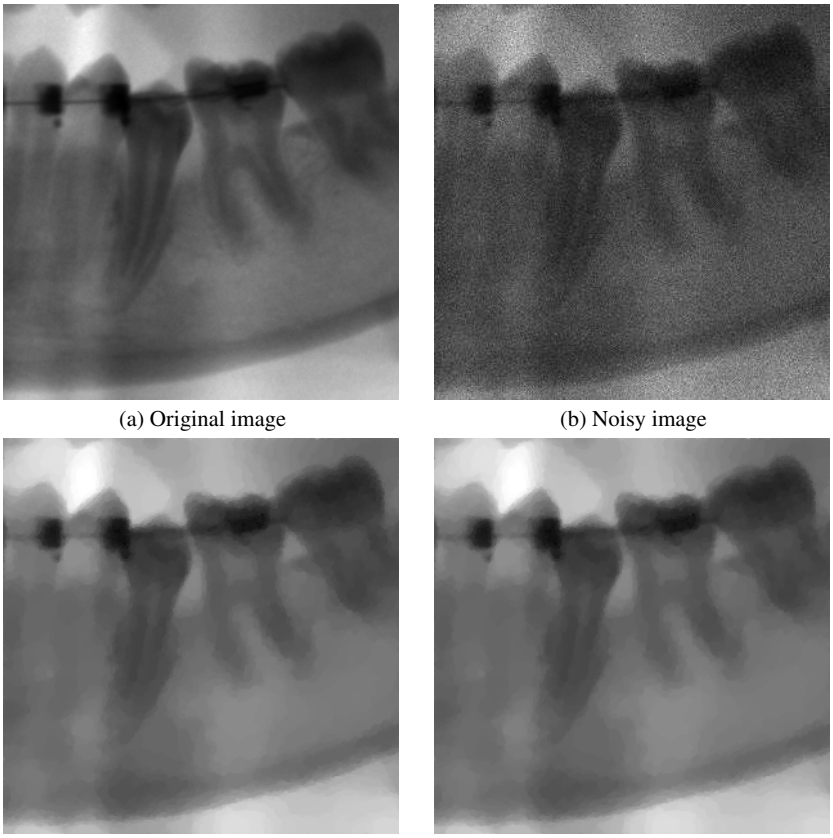

**Fig. 2** Reconstructions at the level $\mu = 10^{-4}$: lineouts of row 128 of the image LCR-0.2. Left panel: superposition of the row 128 of the original image and of the noisy image. Right panel: reconstruction by Algorithm 1 (706 *it* and *time*=20.2 seconds).



(a) Original image (b) Noisy image

(c) Algorithm 1 (d) PIDSplit+ $\gamma = \frac{5}{\beta}$

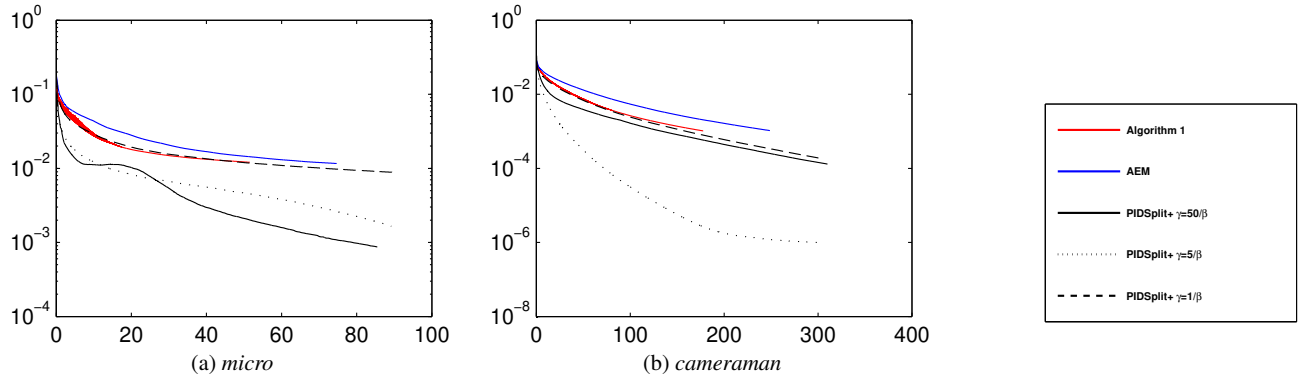
**Fig. 3** Test problem AIR, $\beta = 0.05$, $\mu = 10^{-4}$. Upper left panel: original image. Upper right panel: noisy image. Bottom left panel: Algorithm 1 reconstruction. Bottom right panel: PIDSplit+ reconstruction with $\gamma = \frac{5}{\beta}$.



(a) Original image (b) Noisy image

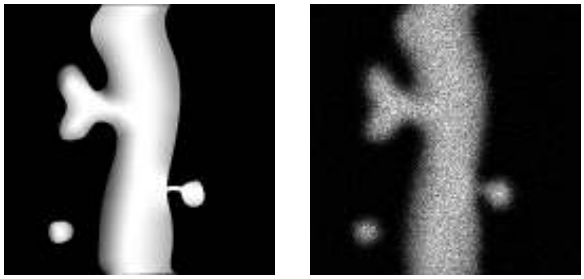
(c) Algorithm 1 (d) PIDSplit+ $\gamma = \frac{0.5}{\beta}$

**Fig. 4** Test problem DR $\beta = 0.27$, $\mu = 10^{-4}$. Upper left panel: original image. Upper right panel: noisy image. Bottom left panel: Algorithm 1 reconstruction. Bottom right panel: PIDSplit+ reconstruction, $\gamma = \frac{0.5}{\beta}$.

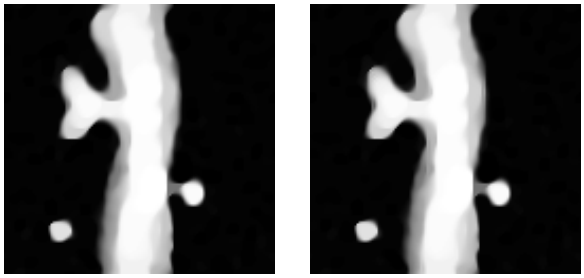



(a) *micro* (b) *cameraman*

**Fig. 5** Deblurring problems: convergence speed of Algorithm 1, AEM, PIDSplit+ (with different value of $\gamma$): plot of $l_2$ norm of the relative error $e^k$ with respect to the ideal solution of the minimization problem versus the computational time. All the methods run for 3000 iterations.



(a) Original image (b) Noisy image



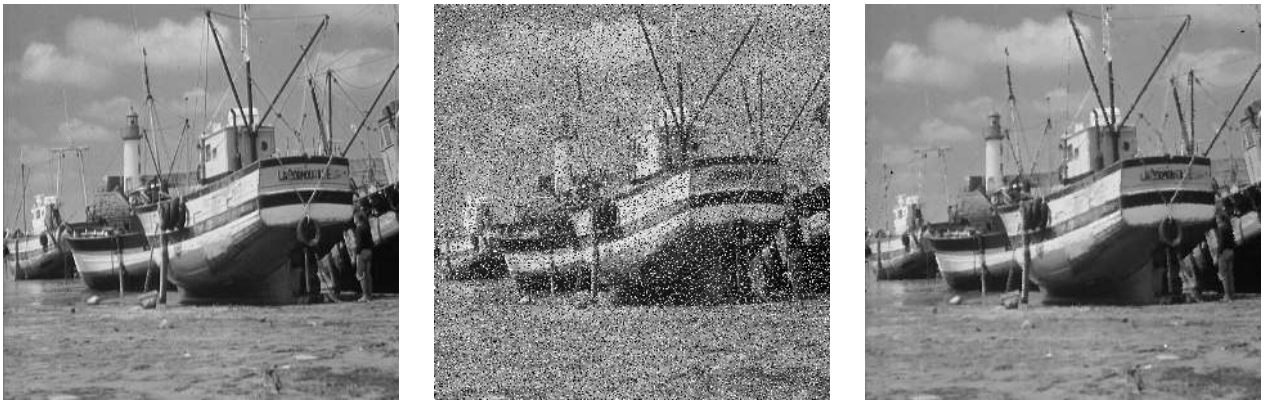
(c) Algorithm 1 (d) PIDSplit+ $\gamma = \frac{50}{\beta}$

**Fig. 6** Test problem *micro*, $\beta = 0.09$, 3000 iterations. Upper left panel: original image. Upper right panel: noisy blurred image. Bottom left panel: Algorithm 1 reconstruction. (*time*= 51.1 seconds). Bottom right panel: PIDSplit+ reconstruction, $\gamma = \frac{50}{\beta}$ (*time*= 85.5 seconds).

(a) Original image (b) Noisy image

(c) Algorithm 1 (d) PIDSplit+ $\gamma = \frac{5}{\beta}$

**Fig. 7** Test problem *cameraman*, $\beta = 0.005$, 3000 iterations. Upper left panel: original image. Upper right noisy blurred image. Bottom left panel: Algorithm 1 reconstruction (*time*= 177.7 seconds). Bottom right panel: PIDSplit+ reconstruction, $\gamma = \frac{5}{\beta}$ (*time*= 306.3 seconds).



(a) Original image (b) Noisy image (c) Reconstruction: Algorithm 2

**Fig. 8** Image denoising in the case of impulse noise: test problem *b*oat, $\beta = 0.65$, 2000 iterations. Left panel: original image. Central panel: noisy image. Left panel: Algorithm 2 reconstruction (*time*= 193 seconds).