Some algorithms for total variation based image restoration

Jean-François Aujol

SOME ALGORITHMS FOR TOTAL VARIATION BASED IMAGE RESTORATION

JEAN-FRANÇOIS AUJOL

CMLA, ENS CACHAN, CNRS, UNIVERSUD
EMAIL: JEAN-FRANCOIS.AUJOL@CMLA.ENS-CACHAN.FR

Abstract. This paper deals with numerical schemes for image restoration. These schemes rely on a duality-based algorithm proposed in 1979 by Bermudez and Moreno, and on general minimization schemes recently developped by Y. Nesterov. Total variation regularization and smoothed total variation regularization are investigated. Algorithms are presented for such regularizations in image restoration. We prove the convergence of all the proposed schemes. We illustrate our study with numerous numerical examples, and we make comparisons between the different schemes presented in the paper. Our experiments are in favor of Bermudez-Moreno approach to get a fast approximation for smoothed total variation regularization, whereas Y. Nesterov scheme seems to perform better for total variation regularization.

Key words. Algorithms, duality, total variation regularization, image restoration.

AMS subject classifications. 68U10, 49M29, 65K10.

1. Introduction. During the last 15 years, total variation regularization has known a great success in image processing [39, 5, 18, 4]. It has been used in many applications such as image restoration, image deblurring, image zooming, image inpainting, ...(see [18, 5] and references therein). In all these approaches, a total variation term $\int |Du|$ is to be minimized in some way. The typical problem is the case of image restoration [39] with the minimization of a functional of the type:

$$\int_\Omega |Du| + \frac{1}{2\mu}\|f-u\|^2 \tag{1.1}$$

$\int |Du|$ stands for the total variation of u [2], and if u is regular it is simply $\int_\Omega |\nabla u|\,dx$. Ω is the domain of definition of the image, a convex open set in $\mathbb{R}^2$. f is the degraded image to restore. The minimizer u of (1.1) is the restored image we want to compute (see for instance [15] for a thorough mathematical analysis of this problem). μ is a weighting parameter which controls the amount of denoising. In the case of zero mean Gaussian noise, μ can can be related to the standard deviation of the noise.

From a numerical point of view, total variation is not straightforward to minimize, since it is not differentiable in zero. A first approach is to regularize it, and instead to consider a term as $\int \sqrt{\beta^2+|\nabla u|^2}\,dx$. We will refer to this choice as smoothed total variation regularization:

$$\int_\Omega \sqrt{\beta^2+|\nabla u|^2}\,dx + \frac{1}{2\mu}\|f-u\|^2 \tag{1.2}$$

The classical approach is then to use the associated Euler-Lagrange equation to compute the solution. Fixed step gradient descent [39], or later quasi-Newton methods [15, 25, 1, 19, 34, 35] have been proposed for instance (see [18, 5] and references therein). Iterative methods have proved successful [9, 24, 7]. A projected-subgradient method can be found in [21].

Ideas from duality have also been proposed: first by Chan and Golub [16], later by A. Chambolle in [13, 14], and then generalized in [22]. Second order cone programming ideas and interior point methods have proved interesting approaches [29, 28]. Recently, it has been shown that graph cuts based algorithms could also be used [14, 23]. Finally, let us notice that it is shown in [41] that Nesterov's scheme [32] provides fast algorithms both for minimizing (1.1) and (1.2).

In this paper, we revisit Chambolle's projection algorithm. We show that a modification of Chambolle projection algorithm, recently suggested in [14], can be seen as a particular instance of a more general algorithm proposed almost 30 years ago by Bermudez and Moreno [8]. It is in fact the classical Uzawa algorithm [20] applied to problem (1.1). We then apply the approach of Bermudez and Moreno to smoothed total variation regularization: this gives a new fast algorithm to minimize functionals such as (1.2). We also prove the convergence of this new scheme. We recall a general class of efficient minimization algorithms introduced by Y. Nesterov in [32]. It has been proved in [41] that they are indeed very efficient for image restoration. We then explain how a recent improvement of these algorithms in [33] can be applied for image restoration. We give some numerical examples of all the schemes introduced in this paper. Our experiments are in favor of Bermudez-Moreno approach to get a fast approximation for smoothed total variation regularization, whereas Nesterov scheme seems to perform better for total variation regularization. Notice that to get a higly accurate solution for smoothed total variation regularization, Nesterov's scheme seems also to be the best choice. However, such an accuracy is not necessary for image restoration.

The organization of the paper is the following. In Section 2, we recall Bermudez-Moreno algorithm [8]. We show how it can be applied to total variation regularization in Section 3. In Section 4, we explain the relations between this scheme and Chambolle's projection algorithm [13], and we give some numerical comparisons. In Section 5, we detail how Bermudez-Moreno algorithm can be applied to smoothed total variation based image restoration. In Section 6, we recall a general class of minimization algorithms introduced by Y. Nesterov in [32]. These algorithms have proved very efficient in [41] for solving image processing problems. We then explain how a recent improvement of these algorithms in [33] can be applied for image restoration. In Section 7 we make some comparisons between the different schemes presented in this paper. Appendix A details the proof of convergence of Bermudez-Moreno algorithm.

2. Bermudez-Moreno algorithm. In this section we present the algorithm proposed by Bermudez and Moreno in [8]. This is a general minimization algorithm. We will then show in the next sections how it can be applied to classical image processing problems. We follow here the presentation of [8, 26]. The general minimization problem considered is the following:

$$\inf_{z\in V}\left\{\frac{1}{2}\langle Az,z\rangle-\langle g,z\rangle+\psi(z)\right\} \tag{2.1}$$

with V Hilbert space, ψ a proper convex lower semi continuous (l.s.c.) function defined on V:

$$\psi=\phi o\Lambda_E^{-1}oB^* \tag{2.2}$$

with E Hilbert space, B bounded linear operator, $B:E\to V'$, $\Lambda_E:E\to E'$ (canonical isomorphism from E onto its dual), $B^*:V\to E'$, $\Lambda_E^{-1}:E'\to E$, $\phi:E\to$

$\mathbb{R}$. We recall that if H is a convex function, we say that it is proper if $H(x) > -\infty$ for all x, and if there exists x_0 such that $H(x_0) < +\infty$. We denote by dom H the set on which $H(x) < +\infty$ [12, 26].

Assumptions on A. In all the paper, we will make the following assumptions on A: A is assumed to be a symmetric coercive operator, i.e. there exists $\alpha > 0$ such that for all z in V:

$$\langle Az, z\rangle_{V',V} \geq \alpha \|z\|_V^2 \tag{2.3}$$

We will also make the three following assumptions on A:

$$\begin{cases} A \text{ is weakly continuous on the finite dimensional subspaces of } V. \\ A \text{ is a monotone operator: } \langle Ay - Az, y - z\rangle \geq 0 \ \forall y, z \in V \\ \text{There exists } z_0 \text{ in dom } \psi \text{ such that: } \frac{\langle Az, z - z_0\rangle + \psi(z)}{\|z\|} \to +\infty \text{ if } \|z\| \to +\infty \end{cases} \tag{2.4}$$

Notice that in the next sections, all these assumptions will indeed be satisfied. In particular, since we will only consider operator A of the type $A = \gamma I$ for some $\gamma > 0$, the technical assumption (2.4) will be trivially verified.

Notations. We use the following notations [37]. If H is a maximal monotone operator, we denote by H_λ its Yosida approximation:

$$H_\lambda = \frac{I - L_\lambda}{\lambda} \quad \text{where} \quad L_\lambda = (I + \lambda H)^{-1} \tag{2.5}$$

Bermudez and Moreno derive their results for $H = \partial\phi - \omega I$. Here we take $\omega = 0$, and we consider the scheme (2.7) with:

$$H = \partial\phi \tag{2.6}$$

Notice that since ϕ is assumed to be a convex proper lower semi continuous function, $\partial\phi$ is a maximal monotone operator [11, 12, 37, 4].

Algorithm. In [8], Bermudez and Moreno propose to use the following algorithm to minimize (2.1). u^0 being arbitrary, consider the iterative scheme:

$$\begin{cases} y^m = A^{-1}(g - Bu^m) \\ u^{m+1} = H_\lambda\left(\Lambda_E^{-1} B^* y^m + \lambda u^m\right) \end{cases} \tag{2.7}$$

They prove the following convergence result (proposition 3.1 in [8]):

THEOREM 2.1. *Assume that A a symmetric coercive operator satisfying (2.3) and (2.4), and that ϕ is a convex proper lower semi continuous function. Assume furthermore that:*

$$0 < \frac{1}{\lambda} < \frac{2\alpha}{\|B^*\|^2} \tag{2.8}$$

Then the sequence (y^m) defined by (2.7) is such that: $\lim_{m\to+\infty} y^m = y$ with y solution of: $g - Ay \in B\partial\phi(\Lambda_E^{-1} B^* y)$, i.e. y unique solution of (2.1). Moreover, $u^m \rightharpoonup u$ in E weak, with: $u \in \partial\phi(\Lambda_E^{-1} B^* y)$.

The proof of Theorem 2.1 is detailed in Appendix A. One of the main interest of Theorem 2.1 is that it is not restricted to the case when ϕ is a support function [26]. However, due to the importance of total variation regularization in image processing, we first consider the case of problem (1.1) in Sections 3 and 4. We will consider the case of problem (1.2) in Section 5, where ϕ is no longer a support function.

3. Application to total variation regularization. In this section, we show how Bermudez-Moreno algorithm (2.7) can be used for total variation regularization. In Section 3.1, we first consider the continuous setting to derive the link with Bermudez-Moreno's work. In Section 3.2, we then consider the discrete case and we propose an efficient algorithm.

3.1. Continuous setting. Let us consider the celebrated ROF model [39]:

$$\inf_{u\in L^2(\Omega)} J(u) + \frac{1}{2\mu}\|f-u\|^2_{L^2(\Omega)} \tag{3.1}$$

Here $J(u)$ is the total variation of u extended to $L^2(\Omega)$ (since in dimension 2, we have $BV(\Omega)\subset L^2(\Omega)$ [2]):

$$J(u) = \begin{cases} \int_\Omega |Du| & \text{if } u\in BV(\Omega) \\ +\infty & \text{otherwise} \end{cases} \tag{3.2}$$

In fact, (3.1) is a particular case of (2.1). Indeed, take $V = L^2(\Omega)$, $E = (L^2(\Omega))^2$, $A = \frac{1}{\mu}I$, $g = \frac{1}{\mu}f$. A is of course coercive with coercivity constant $\alpha = \frac{1}{\mu}$. $J(u) = \psi(u) = \phi(B^*(u))$, and

$$J(u) = \sup_{v\in K} \langle u, \operatorname{div} v\rangle \tag{3.3}$$

Hence ϕ support function of K (closed convex set in $(L^2(\Omega))^2$):

$$K = \left\{ v\in (L^2(\Omega))^2 \ / \ \operatorname{div} v \in L^2(\Omega)\ ,\ \|v\|_\infty \le 1 \text{ with } |v| = \sqrt{v_1^2+v_2^2} \right\} \tag{3.4}$$

We have: $\phi(u) = \sup_{v\in K}\langle u, v\rangle_{L^2(\Omega)}$, $B = -\text{div } = \nabla^*$, and $B^* = \nabla$. Moreover, since ϕ is the support function of K, then $H_\lambda(v)$ is the orthogonal projection of $\frac{v}{\lambda}$ onto K [37, 8] , i.e.: $H_\lambda(v) = P_K\left(\frac{v}{\lambda}\right)$, where

$$P_K(x) = \left(\frac{x_1}{\max\{1,|x|\}}, \frac{x_2}{\max\{1,|x|\}}\right) \tag{3.5}$$

Bermudez-Moreno algorithm (2.7) in this case is therefore: u^0 arbitrary, and:

$$\begin{cases} u^m = f + \mu \operatorname{div} y^m \\ y^{m+1} = P_K\left(y^m + \frac{1}{\lambda}\nabla u^m\right) \end{cases} \tag{3.6}$$

Applying Theorem 2.1, we get the following result:

PROPOSITION 3.1. *If $\lambda > \frac{\mu}{2}\|B^*\|^2$, then the sequence (u^m, y^m) defined by scheme (3.6) is such that $u^m \to u$ and $y^m \to y$ with u solution of (3.1).*

Notice that in this case, algorithm (3.6) is in fact Uzawa algorithm [20] applied to problem (3.1).

3.2. Discrete setting. From now on, and until the end of the paper, we will restrict our attention to the discrete setting. We take here the same notations as in [13]. The image is a two dimension vector of size $N \times N$. We denote by X the Euclidean space $\mathbb{R}^{N\times N}$, and $Y = X \times X$. The space X will be endowed with the scalar product $(u, v) = \sum_{1\leq i,j\leq N} u_{i,j}v_{i,j}$ and the norm $\|u\| = \sqrt{(u, u)}$. To define a discrete total variation, we introduce a discrete version of the gradient operator. If $u \in X$, the gradient ∇u is a vector in Y given by: $(\nabla u)_{i,j} = ((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j})$. with

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \text{ and } (\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

The discrete total variation of u is then defined by:

$$J(u) = \sum_{1\leq i,j\leq N} |(\nabla u)_{i,j}| \tag{3.7}$$

We also introduce a discrete version of the divergence operator. We define it by analogy with the continuous setting by $\text{div} = -\nabla^*$ where ∇^* is the adjoint of ∇: that is, for every $p \in Y$ and $u \in X$, $(-\text{div}\, p, u)_X = (p, \nabla u)_Y$. It is easy to check that:

$$(\text{div}\,(p))_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if i=1} \\ -p^1_{i-1,j} & \text{if i=N} \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if j=1} \\ -p^2_{i,j-1} & \text{if j=N} \end{cases} \tag{3.8}$$

From now on, we will use these discrete operators. *Notice that in all the rest of the paper (except in the appendix), we place ourself in the discrete setting. We will somtimes use continuous notations; however, the reader has to keep in mind that only the discrete case is considered.*

We will use Meyer G space for oscillating patterns [30]:

$$G = \{v \in X \ / \ \exists g \in Y \text{ such that } v = \text{div}\,(g)\} \tag{3.9}$$

and if $v \in G$:

$$\begin{aligned} \|v\|_G = \inf \{\|g\|_\infty \ / \ v = \text{div}\,(g), \\ g = (g^1, g^2) \in Y, |g_{i,j}| = \sqrt{(g^1_{i,j})^2 + (g^2_{i,j})^2}\Big\} \end{aligned} \tag{3.10}$$

where $\|g\|_\infty = \max_{i,j} |g_{i,j}|$. Moreover, we will denote:

$$G_\mu = \{v \in G \ / \ \|v\|_G \leq \mu\} \tag{3.11}$$

With these classical finite differences, we have: $\|\nabla u\|^2 \leq 8\|u\|^2$. Hence $\|\nabla\|^2 = \|B^*\|^2 \leq 8$. And in fact it is possible to show [13] that $\|\nabla\|^2 = \|B^*\|^2 = 8$.

Let us consider new variables:

$$v^m = \frac{u^m}{\mu} \ , \ p^m = y^m \ , \ \tau = \frac{\mu}{\lambda} \tag{3.12}$$

Then we can rewrite (3.6) into: p^0 arbitrary, and

$$\begin{cases} v^m = \frac{f}{\mu} + \text{div}\, p^m \\ p^{m+1} = P_K\left(p^m + \tau \nabla v^m\right) \end{cases} \tag{3.13}$$

FIG. 3.1. *The classical Lenna and cameraman image, and their noisy version (additive zero mean gaussian noise with standard deviation $\sigma = 20$).*

Applying Theorem 2.1, we get the following result:

PROPOSITION 3.2. *If $\tau < \frac{1}{4}$, then the sequence (v^m, p^m) defined by scheme (3.13) is such that $v^m \to v$ and $p^m \to p$ with μv solution of (3.1).*

Notice that (3.13) can be written in a more compact way:

$$p^{m+1} = P_K\left(p^m + \tau\nabla\left(\frac{f}{\mu} + \operatorname{div} p^m\right)\right) \tag{3.14}$$

3.3. Numerical examples. We show here some numerical experiments with (3.13). We will make some comparisons with other existing algorithms in Section 7, and at this point we will just mention that (3.13) is indeed a fast algorithm to solve total variation regularization problem in image processing (less then 3 seconds for a 256×256 image). On Figure 3.1, we display the classical images Lenna and cameraman that we use in this paper to illustrate our study. We also show their noisy versions (degraded by additive zero mean gaussian noise with standard deviation $\sigma = 20$). On Figure 3.2, we show the restoration we get with (3.13). These results have the classical behaviour of total variation based image restoration.

4. Relation with Chambolle projection algorithm. In this section, we explain the link of our scheme (3.13) with a famous projection algorithm proposed by

FIG. 3.2. *Total variation restoration of the noisy images presented on the bottom row of Figure 3.1 with scheme (3.13). In both cases, the Lagrange multiplier is* $\mu = 30$.

A. Chambolle in [13]. We therefore first recall Chambolle's projection algorithm in Section 4.1, and then we compare it with our scheme (3.13) in Section 4.2.

4.1. Total variation minimization as a projection.

Introduction. Since J defined by (3.2) is homogeneous of degree one (i.e. $J(\lambda u) = \lambda J(u)\ \forall u$ and $\lambda > 0$), it is then standard (see [26]) that $J^*(v) = \sup_u(J(u) - \langle u, v\rangle)$ is the indicator function of some closed convex set, which turns out to be the set G_1 defined by (3.11):

$$J^*(v) = \chi_{G_1}(v) = \begin{cases} 0 & \text{if } v \in G_1 \\ +\infty & \text{otherwise} \end{cases} \tag{4.1}$$

This can be checked out easily (see [13] for details). In [13], A. Chambolle proposes a nonlinear projection algorithm to minimize the ROF model. The problem is:

$$\inf_{u\in X}\left(J(u) + \frac{1}{2\mu}\|f-u\|_X^2\right) \tag{4.2}$$

The following result is shown:

PROPOSITION 4.1. *The solution of (4.2) is given by:*

$$u = f - P_{G_\mu}(f) \tag{4.3}$$

where P is the orthogonal projector on G_μ (defined by (3.11)).

Algorithm. [13] gives an algorithm to compute $P_{G_\mu}(f)$. It indeed amounts to finding:

$$\min\left\{\|\mu\mathrm{div}\,(p) - f\|_X^2 : p \ / \ |p_{i,j}| \le 1\ \forall i,j = 1,\dots,N\right\} \tag{4.4}$$

This problem can be solved by a fixed point method: $p^0 = 0$, and

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(\mathrm{div}\,(p^n) - f/\mu))_{i,j}}{1 + \tau|(\nabla(\mathrm{div}\,(p^n) - f/\mu))_{i,j}|} \tag{4.5}$$

In [13] is given a sufficient condition ensuring the convergence of the algorithm:

THEOREM 4.2. *Assume that the parameter τ in (4.5) verifies $\tau < 1/8$. Then $\mu\mathrm{div}\,(p^n)$ converges to $P_{G_\mu}(f)$ as $n \to +\infty$.*

In practice, convergence of the above algorithm is generally observed as long as $\tau < 1/4$. An extension of this algorithm to color images has been proposed in [10]. The case of more general Hilbert space has been considered in [6].

4.2. A modified projection algorithm. In [14], A. Chambolle has proposed a modification of his projection algorithm developed in [13], whose algorithm (4.5) can be rewritten as:

$$\left\{ \begin{array}{l} v^m = \frac{f}{\mu} + \operatorname{div} p^m \\ p_{i,j}^{m+1} = \frac{p_{i,j}^m + \tau(\nabla v^m)_{i,j}}{1+\tau|(\nabla v^m)_{i,j}|} \end{array} \right. \tag{4.6}$$

and μv^m converges to the solution of (4.2). Instead of using (4.6), he suggests in [14] to use a simple gradient descent/retroprojection method:

$$\left\{ \begin{array}{l} v^m = \frac{f}{\mu} + \operatorname{div} p^m \\ p_{i,j}^{m+1} = \frac{p_{i,j}^m + \tau(\nabla v^m)_{i,j}}{\max\left\{1, |p_{i,j}^n + \tau(\nabla v^m)_{i,j}|\right\}} \end{array} \right. \tag{4.7}$$

And this last equation is exactly scheme (3.13). In [14], A. Chambolle has proved the stability of (4.7). Application of basic results about the projected gradient algorithm [38] shows that in fact (4.7) is convergent provided $\tau < 1/4$. The result of Proposition 3.2 therefore confirms the condition $\tau < 1/4$. Moreover, as we will see in the next section, the general algorithm (2.7) proposed by Bermudez and Moreno [8] can be of interest to other image restoration problems, such as smoothed total variation regularization based ones (1.2).

Numerical comparisons of all these schemes ((3.13), (4.6)) will be discussed in Section 7.2.

5. Smoothed total variation regularization. In this section, we consider the following problem:

$$\inf_u \int_\Omega \sqrt{\beta^2 + |\nabla u|^2} + \frac{1}{2\mu}\|f-u\|_2^2 \tag{5.1}$$

We refer to this problem as the smoothed total variation based regularization problem. For small values of β it can be seen as an approximation of (3.1). This type of regularization is very common in image processing (see [5, 18] and references therein). Compared to total variation regularization, it has the advantage of being a smooth regularization. And compared to stronger regularization such as $\|\nabla u\|^2$, it has the advantage of not eroding too much the edges of the image.

In Section 5.1, we explain how Bermudez-Moreno algorithm (2.7) can be used to solve this problem. The new algorithm we propose has a fixed point iteration step. We show the convergence of this fixed point iteration in Section 5.3. We will show some numerical examples with this new scheme in Section 7.1.

5.1. Presentation of the scheme. Let us denote by

$$\phi_\beta(\xi) = \int \sqrt{\beta^2 + |\xi|^2} \tag{5.2}$$

We have

$$\partial\phi_\beta(\xi) = \frac{\xi}{\sqrt{\beta^2 + |\xi|^2}} \tag{5.3}$$

Let us consider the following scheme:

$$\begin{cases} u^m = f + \mu \operatorname{div} y^m \\ y^{m+1} = \frac{I-(I+\lambda\partial\phi_\beta)^{-1}}{\lambda}\left(\nabla u^m + \lambda y^m\right) \end{cases} \tag{5.4}$$

Applying Theorem 2.1, we get:

PROPOSITION 5.1. *If $\lambda > 4\mu$, then the sequence (u^m, y^m) defined by scheme (5.4) is such that $u^m \to u$ and $y^m \to y$ with u solution of (5.1).*

The second equation of (5.4) implies:

$$\lambda y^{m+1} = \nabla u^m + \lambda y^m - (I + \lambda\partial\phi_\beta)^{-1}\left(\nabla u^m + \lambda y^m\right) \tag{5.5}$$

As in the total variation case, let us set:

$$v^m = \frac{u^m}{\mu} \text{ and } \tau = \frac{\mu}{\lambda} \text{ and } y^m = p^m \tag{5.6}$$

(5.4) becomes:

$$\begin{cases} v^m = \frac{f}{\mu} + \operatorname{div} p^m \\ (I + \lambda\partial\phi_\beta)\left(\lambda(\tau\nabla v^m + p^m - v^{m+1})\right) = \lambda(\tau\nabla v^m + p^m) \end{cases} \tag{5.7}$$

Let us set:

$$w^{m+1} = \tau\nabla v^m + p^m - p^{m+1} \tag{5.8}$$

From the second line of (5.7), we get:

$$w^{m+1} + \partial\phi_\beta\left(\lambda(w^{m+1}\right) = \tau\nabla v^m + p^m \tag{5.9}$$

But

$$\partial\phi_\beta\left(\lambda w^{m+1}\right) = \frac{\lambda w^{m+1}}{\sqrt{\beta^2 + |\lambda w^{m+1}|^2}} = \frac{w^{m+1}}{\sqrt{\frac{\beta^2}{\lambda^2} + |w^{m+1}|^2}} \tag{5.10}$$

We thus get from (5.9)

$$w^{m+1} + \frac{w^{m+1}}{\sqrt{\frac{\beta^2\tau^2}{\mu^2} + |w^{m+1}|^2}} = \tau\nabla v^m + p^m \tag{5.11}$$

Using the notations $\gamma = \frac{\beta\tau}{\mu}$, and $C^m = \tau\nabla v^m + p^m$, the previous equation becomes:

$$w^{m+1}\left(1 + \frac{1}{\sqrt{\gamma^2 + |w^{m+1}|^2}}\right) = C^m \tag{5.12}$$

(5.12) is easily solved with a fixed point iteration. Indeed we have the following result:

PROPOSITION 5.2. *Consider the sequence* $x_0 = w^m$*:*

$$x_{k+1} = C^m \left(\frac{\sqrt{\gamma^2 + |x_k|^2}}{1 + \sqrt{\gamma^2 + |x_k|^2}} \right) \tag{5.13}$$

Then $x_k \to w^{m+1}$ *as* $k \to +\infty$*.*

Bermudez and Moreno algorithm has already been used for smoothed total variation based restoration in [3]. The authors of [3] use a different approach then in this paper. To solve (5.12), they take the square of both sides of (5.12), and they use a Newton method to compute $|w^{m+1}|$. They then compute w^{m+1} with (5.12). But with such an approach, the authors of [3] report poor numerical results. We also tried this approach, and we have seen the same poor results as in [3]. We therefore advocate the use of the fixed point algorithm proposed here to solve (5.12), which we prove to converge without further assumption (notice that another alternative would be to solve directly (5.12) with Newton method). The final scheme to solve (5.1) is thus:

$$\begin{cases} v^m = \frac{f}{\mu} + \operatorname{div} p^m \\ w^{m+1} = \left(1 + \frac{1}{\sqrt{\frac{\beta^2\tau^2}{\mu^2} + |w^{m+1}|^2}} \right)^{-1} (\tau \nabla v^m + p^m) \\ p^{m+1} = \tau \nabla v^m + p^m - w^{m+1} \end{cases} \tag{5.14}$$

The second equation is solved with a fixed point iteration (5.13). We will see that in practice, a single iteration is enough, and thus the second line of (5.14) reduces to:

$$w^{m+1} = \left(1 + \frac{1}{\sqrt{\frac{\beta^2\tau^2}{\mu^2} + |w^m|^2}} \right)^{-1} (\tau \nabla v^m + p^m) \tag{5.15}$$

Applying theorem 2.1, we have the following convergence result:

PROPOSITION 5.3. *If* $\tau < \frac{1}{4}$*, then the sequence* (v^m, w^m, p^m) *defined by scheme (5.14) is such that* $v^m \to v$*,* $w^m \to w$*, and* $p^m \to p$ *with* μv *solution of (5.1).*

5.2. Interpretation of scheme (5.14). One first needs to remember that we are interested in solving problem (5.1). Using the change of notation $v = u/\mu$, solving (5.1) is equivalent to solving:

$$\inf_v \int \sqrt{\frac{\beta^2}{\mu^2} + |\nabla v|^2} + \frac{1}{2} \left\| \frac{f}{u} - v \right\|^2 \tag{5.16}$$

The associated Euler-equation is:

$$0 = v - \frac{f}{u} - \operatorname{div} \left(\frac{\nabla v}{\sqrt{\frac{\beta^2}{\mu^2} + |\nabla v|^2}} \right) \tag{5.17}$$

The most classical methods to solve this equation are the fixed step gradient descent as in [39], and the quasi-Newton method (which can be seen also as semi-quadratic regularization) as for instance in [15, 25, 1, 19, 17, 35]. The idea of the

quasi-Newton method is to linearize the non-linear term in the above equation, and to consider an iterative scheme of the type:

$$0 = v_{m+1} - \frac{f}{u} - \operatorname{div}\left(\frac{\nabla v_{m+1}}{\sqrt{\frac{\beta^2}{\mu^2} + |\nabla v_m|^2}}\right) \tag{5.18}$$

Here, we propose a different iterative scheme to solve (5.17).

$$0 = v_m - \frac{f}{u} - \operatorname{div} p^m \tag{5.19}$$

with

$$p^m = \frac{\nabla z_m}{\sqrt{\frac{\beta^2}{\mu^2} + |\nabla z_m|^2}} \tag{5.20}$$

In the limit, we would like to have $z^m \to \nabla v$. To update p^m, we use the following equation:

$$p^{m+1} = p^m + \tau\left(\nabla v^m - z^{m+1}\right) \tag{5.21}$$

If (p^m) converges, then $v^m \to v$ with (5.19), and $z^m \to \nabla v$ with (5.21) as $m \to +\infty$. The system of equations (5.19)-(5.20)-(5.21) can be rewritten into:

$$\begin{cases} v^m = \frac{f}{\mu} + \operatorname{div} p^m \\ z^{m+1}\left(\tau + \frac{1}{\sqrt{\frac{\beta^2}{\mu^2} + |z^{m+1}|^2}}\right) = \tau \nabla v^m + p^m \\ p^{m+1} = p^m + \tau(\nabla v^m - z^{m+1}) \end{cases} \tag{5.22}$$

If we make the change of variable $w^m = z^m/\tau$, then scheme (5.22) is exactly (5.14), i.e. Bermudez-Moreno algorithm for solving problem (5.1).

5.3. Convergence of the fixed point iteration. We give here the proof of proposition 5.2. The proof relies on Weizfeld method [40, 17, 27]. We adopt here the presentation of [17] for Weizfeld method. Let us first introduce some notations. We consider the following functional:

$$F(u) = \frac{1}{2}\|u - C\|^2 + \|(\gamma^2 + |u|^2)^{1/4}\|^2 \tag{5.23}$$

We have:

$$\nabla F(u) = u - C + \frac{u}{\sqrt{\gamma^2 + |u|^2}} \tag{5.24}$$

Let us define:

$$\mathcal{A}(u) = Id + \frac{Id}{\sqrt{\gamma^2 + |u|^2}} \tag{5.25}$$

Notice that $u \to \mathcal{A}(u)$ is continuous, and that $\lambda_{\min}(\mathcal{A}(u)) \geq 1$, where $\lambda_{\min}(M)$ stands for the smallest eigenvalue of M. Let us finally define:

$$G(v,u) = F(u) + \langle v-u, \nabla F(u)\rangle + \frac{1}{2}\langle v-u, \mathcal{A}(u)(v-u)\rangle \tag{5.26}$$

In fact, this last functional defines a general Weizfeld method for the problem:

$$\inf_u F(u) \tag{5.27}$$

Notice that since F is strictly convex and coercice, there exists a unique u solution of (5.27), and u is the solution of:

$$\nabla F(u) = u\left(1 + \frac{1}{\sqrt{\gamma^2 + |u|^2}}\right) - C = 0 \tag{5.28}$$

We now define the iteration of Weizfeld method:

$$u_{m+1} = \operatorname*{argmin}_v G(v, u_m) \tag{5.29}$$

Since G is strictly convex and coercice, there exists a unique u_{m+1} solution of (5.29). It satisfies the Euler-Lagrange equation:

$$\nabla F(u_m) + \langle \mathcal{A}(u_m)(u_{m+1} - u_m)\rangle = 0 \tag{5.30}$$

i.e.:

$$u_{m+1}\left(1 + \frac{1}{\sqrt{\gamma^2 + |u_m|^2}}\right) = C \tag{5.31}$$

which is precisely iteration (5.13).

PROPOSITION 5.4. *If u is fixed, then for all v we have: $G(v,u) - F(v) \geq 0$.*

Proof.

A standard computation leads to:

$$\begin{aligned} G(v,u) - F(v) &= \langle u - v, \frac{-1}{2}\frac{u+v}{\sqrt{\gamma^2+|u|^2}}\rangle + \int \left(\sqrt{\gamma^2+|u|^2} - \sqrt{\gamma^2+|v|^2}\right) \\ &= \int \frac{-1}{2}\frac{|u|^2-|v|^2}{\sqrt{\gamma^2+|u|^2}} + \int \left(\sqrt{\gamma^2+|u|^2} - \sqrt{\gamma^2+|v|^2}\right) \end{aligned}$$

Using the notation $a = \sqrt{\gamma^2+|u|^2}$ and $b = \sqrt{\gamma^2+|v|^2}$, we get:

$$G(v,u) - F(v) = \int \left(\frac{-1}{2}\frac{a-b}{a} + a - b\right) = \int \frac{(a+b)^2}{2a} \geq 0 \tag{5.32}$$

□

The following lemma holds:

LEMMA 5.5. *We have for all m:*

$$F(u_{m+1}) \leq F(u_m) \tag{5.33}$$

and

$$\lim_{m\to+\infty} \|u_{m+1} - u_m\| = 0 \tag{5.34}$$

Proof. From Proposition 5.4, we have $F(u_{m+1}) \leq G(u_{m+1}, u_m)$. But from (5.29), we get $G(u_{m+1}, u_m) \leq G(u_m, u_m) = F(u_m)$. We thus deduce inequality (5.33).

We now concentrate on proving (5.34). From Proposition 5.4, we have:

$$\begin{aligned} F(u_{m+1}) \leq G(u_{m+1}, u_m) &= F(u_m) + \langle u_{m+1} - u_m, \nabla F(u_m)\rangle \\ &\quad + \frac{1}{2}\langle u_{m+1} - u_m, \mathcal{A}(u_m)(u_{m+1} - u_m)\rangle \\ &= F(u_m) - \frac{1}{2}\langle u_{m+1} - u_m, \mathcal{A}(u_m)(u_{m+1} - u_m)\rangle \end{aligned}$$

We thus deduce that (since $\lambda_{\min}(\mathcal{A}(u)) \geq 1$):

$$\begin{aligned} \frac{1}{2}\|u_{m+1} - u_m\|^2 &\leq \frac{1}{2}\langle u_{m+1} - u_m, \nabla F(u_m)\rangle + \frac{1}{2}\langle u_{m+1} - u_m, \mathcal{A}(u_m)(u_{m+1} - u_m)\rangle \\ &\leq F(u_m) - F(u_{m+1}) \end{aligned}$$

We finally get that:

$$0 \leq \|u_{m+1} - u_m\|^2 \leq \sqrt{2(F(u_m) - F(u_{m+1}))} \tag{5.35}$$

We have just seen before that $F(u_m)$ is a positive monotone decreasing sequence. Hence $F(u_m)$ is a convergent sequence, and in particular $F(u_m) - F(u_{m+1}) \to 0$, which concludes the proof. □

We are now in position to prove the convergence of the fixed point iteration as stated in Proposition 5.2:

Proof. From (5.31), one sees that u_m is uniformly bounded. Therefore, up to a subsequence, u_m converges to some v. Moreover, from Lemma 5.5, we see that u_{m+1} also converges to v. Passing to the limit in (5.31), we see that $v = u$ where u is the unique minimizer of (5.27). We conclude that the whole sequence u_m goes to u. □

We end this section by stating a result about the convergence rate of the fixed point algorithm (5.13). We denote by $\tilde{u}$ the solution of Problem (5.27). We use the following notations:

$$\gamma_m = \frac{G(\tilde{u}, u_m) - F(\tilde{u})}{\frac{1}{2}(\tilde{u} - u_m, \mathcal{A}(u_m)(\tilde{u} - u_m))} \tag{5.36}$$

and

$$\eta = 1 - \lambda_{\min}(\mathcal{A}(\tilde{u})^{-1}\nabla^2 F(\tilde{u})) \tag{5.37}$$

Proposition 5.6.

1. $F(u_{m+1}) - F(\tilde{u}) \leq \gamma_m(F(u_m) - F(\tilde{u}))$.
2. $\eta < 1$ *and* $0 \leq \gamma_m \leq \eta$, *for* t *sufficiently large. In particular,* $F(u_m)$ *has a linear convergence rate of at most* η.
3. u_m *is* r*-linearly convergent with a convergent rate of at most* $\sqrt{\eta}$.

Proof. We refer the interested reader to the proof of Theorem 6.1 in [17]. □

6. Nesterov algorithms. We first recall Nesterov scheme in Section 6.1. It has been proved to be very efficient in [41], and we have therefore decided to use it as a reference algorithm. In Section 6.2, we use an improvement of these schemes as recently introduced by Y. Nesterov in [33] to propose new efficient schemes. In Section 6.3, we present another variation also based on results of [33].

6.1. Nesterov scheme. In [32, 31], Y. Nesterov proposes efficient schemes to minimize functionals such as (3.1) or (5.1). We follow here the presentation of [41]. We consider the following minimization problem:

$$\inf_{u\in Q} E(u) \tag{6.1}$$

where E is a convex Lipshitz differentiable function, and Q a convex closed set. We denote by $\tilde{u}$ a solution of (6.1). For this type of problem, it can be shown that no algorithm (only using the values and gradients of E) has a better rate of convergence than $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ uniformly on all problems of the form (6.1). Moreover, in [32] is given an $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ algorithm for solving problem (6.1):

1. Set $k=-1$, $v_{-1}=0$, $x_{-1}\in Q$, L Lipshitz constant of ∇E.
2. Set $k=k+1$, and compute $\eta_k=\nabla E(x_k)$.
3. Set $y_k=\text{argmin}_{y\in Q}\left(\langle\eta_k, y-x_k\rangle+\frac{1}{2}L\|y-x_k\|^2\right)$.
4. Set $v_k=v_{k-1}+\frac{k+1}{2}\eta_k$.
5. Set $z_k=\text{argmin}_{y\in Q}\left(\frac{L}{\sigma}d(x)+\langle v_k, z\rangle\right)$.
6. Set $x_{k+1}=\frac{2}{k+3}z_k+\frac{k+1}{k+2}y_k$.

PROPOSITION 6.1. *[32] The previous algorithm ensures that :*

$$0\leq E(y_k)-E(\tilde{u})\leq\frac{4Ld(\tilde{u})}{\sigma(k+1)(k+2)} \tag{6.2}$$

At step 3, $\|.\|$ stands for any norm. At step 5, d is any convex function satisfying $d(x)\geq\frac{\sigma}{2}\|x-x_0\|^2$ for some x_0 in Q.

Primal Nesterov algorithm. For $\beta>0$, we remind the reader that we set $\phi_\beta(u)=\int\sqrt{\beta^2+|\nabla u|^2}$. Nesterov algorithm can be used to solve the following problem:

$$\inf_{u\in K_\alpha}\phi_\beta(f+u) \tag{6.3}$$

where $K_\alpha=\left\{x\in L^2\ /\ \|x\|_2\leq\alpha\right\}$.

This problem is equivalent to (5.1) (see [15] for a complete analysis). The advantage of formulation (6.3) is that Nesterov's scheme can directly be applied. See [41] page 13 for a detailed implementation of this algorithm. We will refer to it as the *primal Nesterov algorithm*. We just give here the sketch of the algorithm:

1. Set $k=-1$, $v_{-1}=0$, $x_{-1}=0$, $L=\|\text{div}\,\|^2/\beta=8/\beta$.
2. Set $k=k+1$, and compute $\eta_k=-\text{div}\left(\frac{\nabla(x_k+f)}{\sqrt{\beta^2+|\nabla(x_k+f)|^2}}\right)$.
3. Set $y_k=P_{K_\alpha}(x_k-\eta_k/L)$, with $K_\alpha=\left\{x\in L^2\ /\ \|x\|_2\leq\alpha\right\}$.
4. Set $v_k=v_{k-1}+\frac{k+1}{2}\eta_k$.
5. Set $z_k=P_{K_\alpha}(-v_k/L)$.
6. Set $x_{k+1}=\frac{2}{k+3}z_k+\frac{k+1}{k+2}y_k$.
7. The output of the algorithm is: $u=y_{\text{lim}}+f$.

P_{K_α} is the orthogonal projection over K_α.

Dual Nesterov algorithm. Of course, due to the non-differentiability in zero of the total variation, Nesterov scheme cannot be applied directly to (3.1). The basic idea is to apply Nesterov's scheme to the dual version of (3.1), that is to: $\inf_{f-u\in G_\mu} \frac{1}{2}\|u\|^2$, where G_μ is given by (3.11), i.e.:

$$\inf_{q\in K} E(q) \tag{6.4}$$

where $E(q) = \frac{1}{2}\|f - \mu\mathrm{div}\, q\|^2$ and $K = \left\{x \in L^2 \times L^2 \ / \ \|x\| \leq 1\right\}$. If we denote by $\tilde{u}$ the solution of (3.1), and by $\tilde{q}$ the solution of (6.4), we have $\tilde{u} = f - \mu\mathrm{div}\,\tilde{q}$.

See [41] page 24 for a detailed implementation of this algorithm. We will refer to it as the *dual Nesterov algorithm*. We just give here the sketch of the algorithm:

1. Set $k = -1$, $v_{-1} = 0$, $x_{-1} = 0$, $L = \mu\|\mathrm{div}\,\|^2 = 8\mu$.
2. Set $k = k + 1$, and compute $\eta_k = -\nabla\left(f - \mu\mathrm{div}\,(x_k)\right)$.
3. Set $y_k = P_K(x_k - \eta_k/L)$, with $K = \left\{x \in L^2 \times L^2 \ / \ \|x\| \leq 1\right\}$.
4. Set $v_k = v_{k-1} + \frac{k+1}{2}\eta_k$.
5. Set $z_k = P_K(-v_k/L)$.
6. Set $x_{k+1} = \frac{2}{k+3}z_k + \frac{k+1}{k+2}y_k$.
7. The output of the algorithm is: $u = f - \mu\mathrm{div}\,(y_{\lim})$.

Notice that in the *dual Nesterov algorithm*, the set K is included in $L^2 \times L^2$; whereas in the case of the *primal Nesterov algorithm*, the set K_α is embeded in L^2.

In [41], very good numerical results are reported both for the *primal* and the *dual Nesterov algorithms*. We have therefore decided to use them as reference in the comparisons presented here-after.

6.2. Accelerated Nesterov algorithm. In [33], Y. Nesterov proposes a way to speed up the minimization algorithms introduced in [31]. Consider the general minimization problem

$$\inf_u E(u) + \psi(u) \tag{6.5}$$

We set $\phi(u) = E(u) + \psi(u)$, and:

$$\psi(u) = \chi_Q(u) = \begin{cases} 0 \text{ if } u \in Q \\ +\infty \text{ otherwise.} \end{cases} \tag{6.6}$$

Problem (6.5) is therefore the same as (6.1). As previoulsy, E is a convex Lipshitz differentiable function, and Q a convex closed set. We denote by $\tilde{u}$ a solution of (6.5). Moreover, in [33] is given an efficient algorithm for solving problem (6.5):

- Set $k = 0$, $A_0 = 0$, $v_0 = 0$, $x_0 \in Q$, $L_0 = L$ Lipshitz constant of ∇E, $\psi_0(x) = \frac{1}{2}\|x - x_0\|^2$. Set $\gamma_u > 1$ and $\gamma_d \geq 1$.
- Set $L = L_k$.

REPEAT: Set $a = \frac{1+\sqrt{1+4A_kL}}{2L}$.
Set $y = \frac{A_kx_k + av_k}{A_k + a}$, and compute $T_L(y)$.
If: $\langle \phi'(T_L(y)), y - T_L(y)\rangle < \frac{1}{2L}\|\phi'(T_L(y))\|_2^2$, **then** $L = \gamma_u L$.
UNTIL: $\langle \phi'(T_L(y)), y - T_L(y)\rangle \geq \frac{1}{2L}\|\phi'(T_L(y))\|_2^2$
DEFINE $y_k = y$, $M_k = L$, $a_{k+1} = a$, $A_{k+1} = A_k + a_{k+1}$,
$L_{k+1} = M_k/\gamma_d$, $x_{k+1} = T_{M_k}(y_k)$,
$\psi_{k+1}(x) = \psi_k(x) + a_{k+1}\left(E(x_{k+1}) + \langle \nabla E(x_{k+1}), x - x_{k+1}\rangle + \psi(x)\right)$,
$v_{k+1} = \mathrm{argmin}_x\, \psi_{k+1}(x)$.
Output: the output of the algorithm is $u = x_{\lim}$.

In the above algorithm, we have used the following notations:

$$T_L(y) = \underset{x \in Q}{\operatorname{argmin}}\, m_L(y, x) \tag{6.7}$$

with

$$m_L(y, x) = E(y) + \langle \nabla E(y), x - y \rangle + \frac{L}{2}\|x - y\|^2 + \psi(x) \tag{6.8}$$

Moreover, it is shown in [33] that

$$\phi'(T_L(y)) = L(y - T_L(y)) + \nabla E(T_L(y)) - \nabla E(y) \tag{6.9}$$

The following convergence result is shown in [33]:

PROPOSITION 6.2. *[33] Let L_E the Lipshitz constant of ∇E. Assume that $0 < L_0 \leq L_E$. Then the previous algorithm ensures that :*

$$0 \leq \phi(x_k) - \phi(\tilde{u}) \leq \frac{4\gamma_u L_E \|\tilde{u} - x_0\|^2}{k^2} \tag{6.10}$$

where we recall that $\phi(u) = E(u) + \psi(u)$.

To apply this new algorithm, the only points to check are how to solve (6.7) and how to compute v_k. This is explained by the two following lemmas.

LEMMA 6.3. *The solution of (6.7) is given by:*

$$T_L(y) = P_Q\left(y - \frac{1}{L}\nabla E(y)\right) \tag{6.11}$$

with P_Q orthogonal projection over Q.

Proof. It is easy to see that:

$$m_L(y, x) = C(y) + \frac{L}{2}\left\|x - \left(y - \frac{1}{L}\nabla E(y)\right)\right\|_2^2 + \psi(x) \tag{6.12}$$

where $C(y)$ is a function depending only on y. The result of the lemma follows from the fact that $\psi = \chi_Q$. □

LEMMA 6.4. *$v_k = \operatorname{argmin}_x \psi_k(x)$ is given by*

$$v_k = P_Q\left(x_0 - \sum_{p=1}^{k} a_p \nabla E(x_p)\right) \tag{6.13}$$

with P_Q orthogonal projection over Q.

Proof. Remembering that $\psi_0(x) = \frac{1}{2}\|x - x_0\|^2$, it is easy to see that:

$$\psi_k(x) = C(k) + \sum_{p=1}^{k} a_p \psi(x) + \frac{1}{2}\left\|x - x_0 + \sum_{p=1}^{k} a_p \nabla E(x_p)\right\|_2^2 \tag{6.14}$$

where $C(k)$ is a function depending only on k. The result of the lemma follows from the fact that $\psi = \chi_Q$. □

In practice, as proposed in [33], we use $\gamma_u = \gamma_d = 2$.

Application to problem (6.3). The above algorithm can directly be applied to (6.3), with $E(u) = \phi_\beta(u+f) = \int \sqrt{\beta^2 + |\nabla(f+u)|^2}$, and $\psi(u) = \chi_{K_\alpha}(u)$, where $K_\alpha = \big\{x \in L^2 \ / \ \|x\|_2 \leq \alpha\big\}$. Of course, on has: $\nabla E(u) = -\mathrm{div}\left(\frac{\nabla(f+u)}{\sqrt{\beta^2+|\nabla(f+u)|^2}}\right)$.

One just has to set $x_0 = 0$, $L = \|\mathrm{div}\,\|^2/\beta = 8/\beta$. The solution is given by $f + x_{\lim}$. Notice that here, the projection over $Q = K_\alpha$ is straightforward:
$P_{K_\alpha}(x) = \frac{\alpha x}{\max\{\alpha, \|x\|_2\}}$.

We will refer to this algorithm as the *accelerated primal Nesterov algorithm.*

Application to problem (3.1). The basic idea is to apply the accelerated Nesterov scheme to the dual version of (3.1), that is to (6.4)), i.e.:

$$\inf_q E(q) + \psi(q) \tag{6.15}$$

with $E(q) = \frac{1}{2}\|f - \mu\mathrm{div}\, q\|_2^2$ and $\psi(q) = \chi_K(q)$ with $K = \{g \in L^2 \times L^2, \sqrt{g_1^2 + g_2^2} \leq 1\}$. We therefore have: $\nabla E(q) = \nabla(f - \mu\mathrm{div}\, q)$.

One just has to set $u_0 = 0$, $L = \mu\|\mathrm{div}\,\|^2 = 8\mu$. The solution is given by $f - \mu\mathrm{div}\, x_{\lim}$. Notice that here, the projection over $Q = K$ is straightforward: $P_K(x_1, x_2) = \frac{1}{\max\{1, \|x\|\}}(x_1, x_2))$, with $x = (x_1, x_2)$ and $\|x\| = \sqrt{x_1^2 + x_2^2}$.

We will refer to this algorithm as the *accelerated dual Nesterov algorithm.*

6.3. Variant for the accelerated Nesterov algorithm. In [33], Y. Nesterov proposes in fact a more general algorithm than the one we have presented in Section 6.2. We still consider the general minimization problem

$$\inf_u E(u) + \psi(u) \tag{6.16}$$

But this time ψ is assumed to be a strongly convex function with parameter $\mu_\psi > 0$: in the case when ψ is C^2, it means that the smallest eigenvalue of $\nabla^2\psi$ is $\mu_\psi > 0$.

We set $\phi(u) = E(u) + \psi(u)$. As previoulsy, E is a convex Lipshitz differentiable function We denote by $\tilde{u}$ a solution of (6.16). Moreover, in [33] is given an efficient algorithm for solving problem (6.16): this is exactly the algorithm presented in Section 6.2, the only difference being that in the step **REPEAT**, instead of setting $a = \frac{1+\sqrt{1+4A_kL}}{2L}$, we set:

$$a = \frac{b + \sqrt{b^2 + 4A_k b}}{2} \quad \text{with } b = \frac{1 + \mu_\psi A_k}{L} \tag{6.17}$$

The following convergence result is shown in [33]:

PROPOSITION 6.5. *[33] Let L_E the Lipshitz constant of E, and μ_ψ the convexity parameter of ψ. Assume that $0 < L_0 \leq L_E$. Then the previous algorithm ensures that (6.10) still holds. Moreover, we also have:*

$$0 \leq \phi(x_k) - \phi(\tilde{u}) \leq \gamma_u L_E \|\tilde{u} - x_0\|^2 \left(1 + \sqrt{\frac{\mu_\psi}{8\gamma_u L_E}}\right)^{-2(k-1)} \tag{6.18}$$

Notice that (6.9) still holds in this case. To apply this new algorithm, the only points to check are how to solve (6.7) and how to compute v_k. We particularize the problem, and we consider (5.1), i.e. in (6.16) we take:

$$E(u) = \phi_\beta(u+f) = \int \sqrt{\beta^2 + |\nabla(f+u)|^2} \quad \text{and} \quad \psi(u) = \frac{1}{2\mu}\|u\|^2 \tag{6.19}$$

Notice that we have:

$$L_E = \|\text{div}\,\|^2/\beta = 8/\beta \quad \text{and} \quad \mu_\psi = \frac{1}{\mu} \tag{6.20}$$

The two following lemmas hold.

LEMMA 6.6. *The solution of (6.7) is given by:*

$$T_L(y) = \frac{Ly - \nabla E(y)}{L + \frac{1}{\mu}} \tag{6.21}$$

Proof. It is easy to see that:

$$\nabla_x(m_L(y,x)) = \nabla E(y) + L(x-y) + \frac{x}{\mu} \tag{6.22}$$

□

LEMMA 6.7. $v_k = \text{argmin}_x\, \psi_k(x)$ *is given by:*

$$v_k = \frac{1}{1 + \frac{\sum_{p=1}^k a_p}{\mu}} \left(x_0 - \sum_{p=1}^k a_p \nabla E(x_p) \right) \tag{6.23}$$

Proof. Remembering that $\psi_0(x) = \frac{1}{2}\|x - x_0\|^2$, it is easy to see that:

$$\psi_k(x) = \frac{1}{2}\|x - x_0\|^2 + \sum_{p=1}^k a_p \psi(x) + \sum_{p=1}^k a_p \left(E(x_p) + \langle \nabla E(x_p), x - x_p \rangle \right) \tag{6.24}$$

□

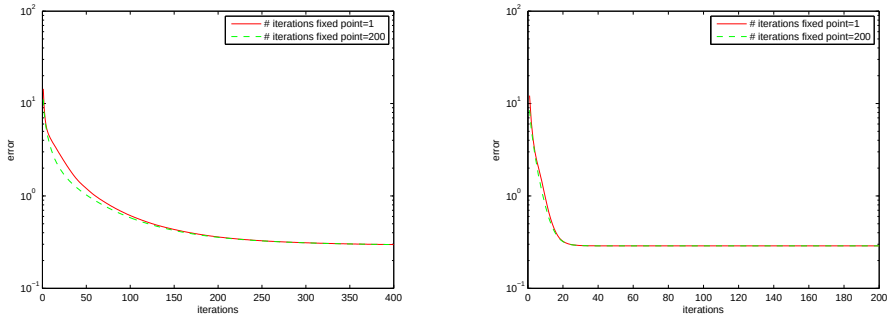
In the next section, we will refer to this algorithm as the *variant of the accelerated primal Nesterov algorithm*. In practice, we take $x_0 = 0$, $\gamma_u = 2$ and $\gamma_d = 2$.

7. Numerical examples. In this section, we present some numerical examples with the schemes introduced in this paper. In Section 7.1, we consider the case of smoothed total variation regularization, and in Section 7.2 we are interested in total variation regularization.

7.1. Smoothed total variation regularization. We illustrate here the efficiency of scheme (5.14) to solve (5.1). This new scheme (5.14) has the advantages of being simple and stable. Moreover, it seems quite fast (less then 2 seconds for a 256×256 image). On Figure 7.1, we show the restoration results we get on the noisy images of Figure 3.1. The curvature parameter β of (5.14) is fixed to 10. As expected, the textures are better preserved with this model than with total variation regularization (compare with Figure 3.2), but the edges are not as sharp.

We now want to see the speed of convergence of (5.14), and how it depends on the number of iterations in (5.13), and on the parameter β. For different values of β, we compute an *ideal* image by running 10 000 iterations of (5.14) with 500 iterations for the fixed point (5.13). We can then compute at each iteration the L^2 error of a computed image with (5.14) and the target *ideal* image. On Figure 7.2, we show the behaviour of the algorithm with respect to the number of iterations for the fixed point iteration, for different values of β. Clearly, it shows that 1 iteration is a very good choice: this will be our choice until the end of the paper. It is also clear that the convergence of (5.14) is much faster for large values of β.

FIG. 7.1. *Smoothed total variation based restoration of the noisy images presented on the bottom row of Figure 3.1 with scheme (5.14) with* $\beta = 10$. *In both cases, the Lagrange multiplier is* $\mu = 30$.



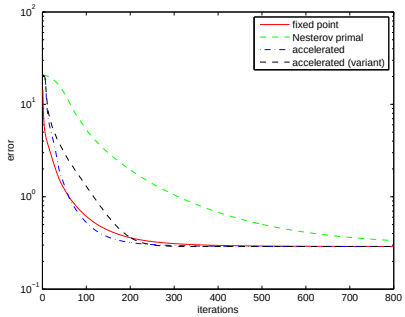
(a) $\beta = 0.1$ (b) $\beta = 10$

FIG. 7.2. *Comparisons of the number of iterations for the fixed point (5.12) in algorithm (5.14): 1 or 200. The* L^2 *error is given with respect to the number of iterations of (5.14) (vertical logarithmic scale). Graph (a) is with* $\beta = 0.1$*: after 60 iterations of (5.14), both errors are the same. Graph (a) is with* $\beta = 10$*: after 10 iterations of (5.14), both errors are the same. We thus advocate to use only 1 iteration for the fixed point iteration (5.13).*

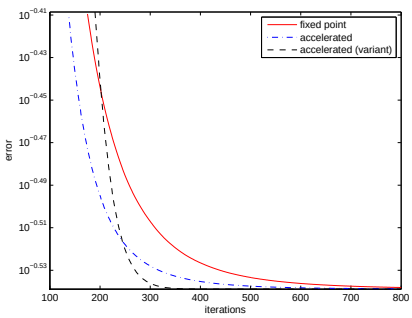
On Figure 7.3, we compare our new algorithm (5.14) with the *primal Nesterov algorithm*, the *accelerated primal Nesterov algorithm*, and the *variant of the accelerated primal Nesterov algorithm* (notice that since algorithm (5.14) uses a fixed point iteration, we refer to it as *fixed point* algorithm in the caption of Figure 7.3). The convergence speed of these three last algorithms depends on the Lipshitz constant of the energy to minimize: the smaller this constant, the faster the method. It thus means here the larger β, the faster the method. Notice that here the images we consider have their range in $[0, 255]$ (while for instance in [41] the images are normalized in $[0, 1]$: this has some impact on the values β proposed).

To make comparisons, we compute the L^2-norm of the difference between the original image and the *ideal* image (obtained by running (5.14) with 10 000 iterations). We then set this L^2-norm as the constraint in the *primal Nesterov algorithm* and the *accelerated primal Nesterov algorithm* . It is to be noticed that such a choice makes a small bias in favor of our scheme (5.14). However, the obtained results are sufficiently convincing to forget this bias.

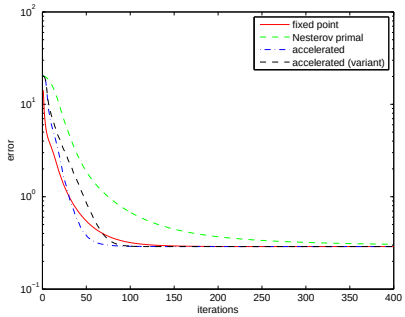
It can be seen that, the larger β, the faster the algorithms. For large values of



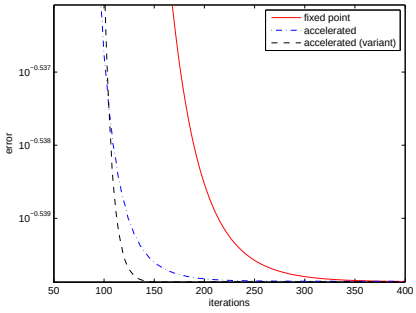
(a) $\beta = 0.1$ (iterations 1-800)



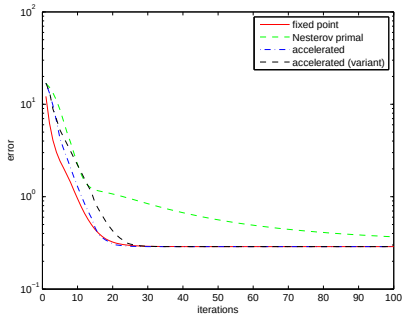
(b) $\beta = 0.1$ (iterations 138-800)



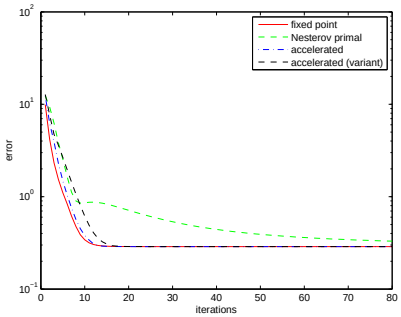
(c) $\beta = 1$ (iterations 1-400)



(d) $\beta = 1$ (iterations 97-400)



(e) $\beta = 10$ (iterations 1-100)



(f) $\beta = 25$ (iterations 0-80)

FIG. 7.3. *Speed of convergence: the* L^2 *error is given with respect to the number of iterations (vertical logarithmic scale). Graph (a) and (b) is with* $\beta = 0.1$*; Graph (c) and (d) with* $\beta = 1$*; Graph (e) with* $\beta = 10$*; Graph (f) with* $\beta = 25$*. The range of the image is between 0 and 255. On graphs (a), (c), (d), (e) and (f), from top to bottom are the speed of convergence of the* fixed point *based algorithm (5.14), the speed of the* primal Nesterov algorithm, *the speed of the* accelerated primal Nesterov algorithm, *the speed of the* variant of the accelerated primal Nesterov algorithm. *On graphs (b) and (d), the* primal Nesterov algorithm *is not shown. Notice that time for 1 iteration of the* primal Nesterov algorithm *is the same as for 1 iteration of the* fixed point *based algorithm (5.14). The* accelerated primal Nesterov algorithm *and its* variant *are between 4 and 5 times slower per iteration. To get a fast approximation, the* fixed point *based algorithm (5.14) seems to be the best choice (the accuracy is good enough for image restoration). To get a higly accurate solution, the* variant of the accelerated Nesterov scheme *seems to be the most efficient.*

β, all the algorithms are fast. However, when β goes down to zero, then scheme (5.14) seems to bring a significant increase in speed of convergence for getting a good approximation. It seems indeed that (5.14) can lead to a good approximation of the minimizer with few iterations. However, when one is interested in getting a very accurate solution, then the *variant of the accelerated primal Nesterov algorithm* seems to be the best choice. This is in accordance with the result of Proposition 6.5. Notice that both scheme (5.14) and the *primal Nesterov algorithm* have almost the same computation time per iteration, while the *accelerated primal Nesterov algorithm* and its *variant* are between 4 and 5 times slower per iteration.

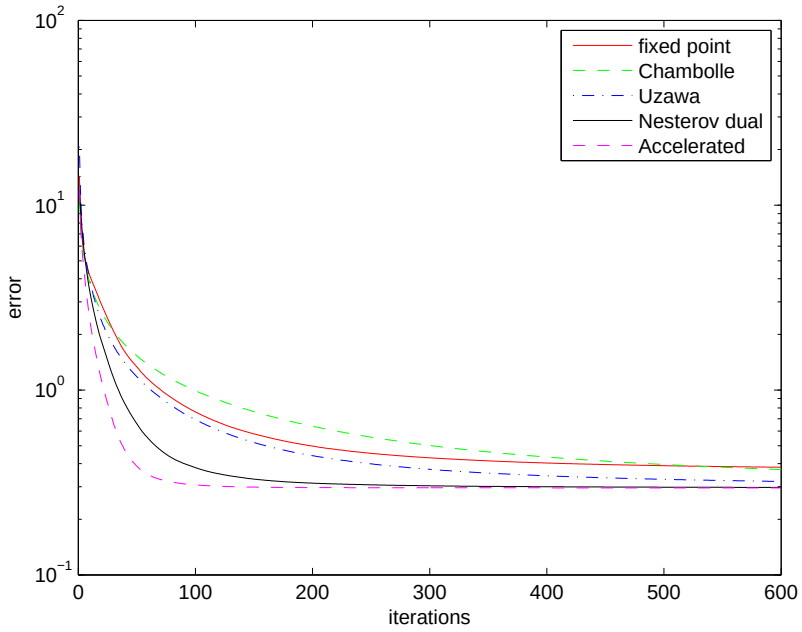
Notice that the quality of the restored image obtained with scheme (5.14) after a few iterations (10 iterations for $\beta = 25$, 20 iterations for $\beta = 10$, 80 iterations for $\beta = 1$, 200 iterations for $\beta = 0.1$) is visually very good. For a restoration purpose, there is no need for the accuracy of the *variant of the accelerated primal Nesterov algorithm*. It is more important to have a fast approximation than a slow and very accurate solution.

7.2. Total variation regularization. Here we consider problem (3.1). We want to compare five different algorithms. The first one is the original projection algorithm of [13]: we refer to it as *Chambolle projection algorithm*. We use $\tau = 0.249$ in (4.6). The second one is the modification of this algorithm as proposed in [14], and which we proved to be Bermudez-Moreno algorithm (3.13) in the case of problem (3.1): since it is Uzawa method [20] applied to problem (3.1), we refer to it as *Uzawa algorithm*. We use $\tau = 0.249$ in (3.13). The third algorithm we use here is the *dual Nesterov algorithm*, as proposed in [41]. Motivated by the results of [32] and [41], we use it as a reference algorithm. The fourth algorithm we use here is the *accelerated dual Nesterov algorithm* of [33]. The fifth algorithm we use is our new scheme (5.14). Since it uses a fixed point algorithm, we refer to it as *fixed point* method.

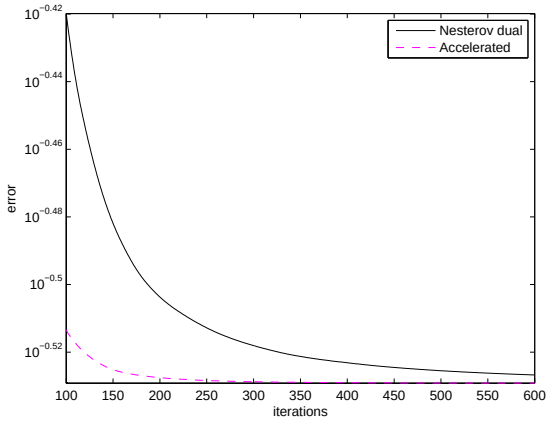
For a given image and a given regularization parameter μ, a reference *ideal* image is computed by running 10 000 iterations with the *dual Nesterov algorithm*. Here, the bias will therefore be in favour of the *dual Nesterov algorithm*. However, we think that the results are convincing enough to forget this bias.

A convergence speed result is presented on Figure 7.4: we give the L^2-norm of $\tilde{u} - u_n$, where u_n is the computed image at iteration n, and $\tilde{u}$ the *ideal* image to obtain. As can be seen on Figure 7.4, the *dual Nesterov* algorithm is faster then *Uzawa* algorithm, which is itself faster than *Chambolle* projection algorithm. The *accelerated dual Nesterov algorithm* seems to be the best choice to get a higly accurate solution. However, 1 iteration with the *accelerated dual Nesterov algorithm* is around 4 times slower than with the *dual Nesterov algorithm*: the *dual Nesterov algorithm* seems thus a good compromise when one is interested in getting a very good approximation. Nevertheless, 1 iteration with the *dual Nesterov algorithm* is around 2 times slower than with *Uzawa*, *Chambolle*, or scheme (5.14) (while all three have the same computation time per iteration). For typical image restoration problems (with Gaussian noise), (3.13) seems 30% faster then (4.6). Algorithm (5.14) seems to be a good alternative when one is only interested in getting an approximation with a small number of iterations

In [41], the authors explain that the *dual Nesterov algorithm* is much faster then the projected gradient method (3.13) (*Uzawa* algorithm) for total variation regularization. We confirm that it is indeed much faster when one is interested in computing an accurate solution. Notice also that in [41], the comparison criterion used is the value of the total variation of the computed image. This is indeed the quantity which



(a) Iterations 1 to 600



(b) Iterations 100 to 600

FIG. 7.4. *Speed of convergence: the* L^2 *norm of the error is given at each iteration (vertical logarithmic scale). (a) gives the speed of convergence for iterations 1 to 600, and (b) for iterations 100 to 600. On graph (a), from top to bottom are the speed of convergence of the* fixed point *based algorithm (5.14) with* $\beta = 0.1$*, the speed of convergence of* Chambolle *projection algorithm (4.6) with* $\tau = 0.249$*, the speed of* Uzawa *scheme (3.13) with* $\tau = 0.249$*, the speed of the* dual Nesterov algorithm*, and the speed of the* accelerated dual Nesterov algorithm*. On graph (b) are only shown the* dual Nesterov algorithm*, and the* accelerated dual Nesterov algorithm*. 1 iteration with the* accelerated dual Nesterov algorithm *is around 4 times slower than with the* dual Nesterov algorithm*. But 1 iteration with the* dual Nesterov algorithm *is itself around 2 times slower than with* Uzawa*,* Chambolle*, or scheme (5.14) (while all three have the same computation time per iteration). To get a higly accurate solution, the* accelerated dual Nesterov *algorithm seems to be the best choice. However, the* dual Nesterov algorithm *seems to be the best compromise when one is only interested in getting a good solution.*

is controlled in Nesterov's approach for solving 3.1 (see Proposition 6.1). Here, the criterion is the L^2 difference of the computed solution for some iteration with the *ideal* solution. Figure 7.4 is surely in favor of the approach developed in [41]. However, the difference during the first iterations is not that large, and thus the projected gradient algorithm (3.13) (*Uzawa algorithm*) can still be considered as a good method when one is only interested in getting an approximation of the solution.

Dual Nesterov algorithm *for solving (5.1).* In view of Figure 7.4, one should be tempted to use the *dual Nesterov algorithm* for solving (5.1). It is easy to compute the dual problem. If we denote by $\tilde{u}$ the solution of (5.1), then we have $\tilde{u} = f - \mu \mathrm{div}\, \tilde{p}$ with $\tilde{p}$ solution of:

$$\inf_{p \in K} \frac{1}{2\mu} \|\mu \mathrm{div}\, p - f\|^2 - \beta \int \sqrt{1 - |p|^2} \tag{7.1}$$

where $K = \left\{ p \in L^2 \times L^2 \ / \ \|p\|_\infty \leq 1 \right\}$. However, the gradient of the functional in (7.1) is not Lipshitz, and we therefore cannot use directly the *dual Nesterov algorithm.*

Practical remark. All the experiments presented in this paper were run with Matlab, on a laptop with a processor at 2.0 GHz.

Appendix A. Proof of convergence of Bermudez-Moreno algorithm. In this section, we follow [8] and [26]. Our goal is to give the reader some intuition on why the result of Theorem 2.1 holds. We remind the reader that we use the notations: $H_\lambda = \frac{I - L_\lambda}{\lambda}$, with $L_\lambda = (I + \lambda H)^{-1}$ and $H = \partial \psi$ with ψ proper convex lower semi continuous function. We will use the next lemma:

LEMMA A.1.

$$\frac{1}{\lambda^2} \left\| L_\lambda(v_1) - L_\lambda(v_2) \right\|^2 + \left\| H_\lambda(v_1) - H_\lambda(v_2) \right\|^2 \leq \frac{1}{\lambda^2} \|v_1 - v_2\|^2 \tag{A.1}$$

Proof. This is an immediate consequence of definitions (2.5). □

Problem (2.1) is related to:

$$\forall z \ , \ \langle Ay, z - y \rangle + \psi(z) - \psi(y) \geq \langle f, z - y \rangle \tag{A.2}$$

The relation is given by the next lemma (whose proof is straightforward (see [26] proposition 2.2 page 37)):

LEMMA A.2. *y is solution of (A.2) if and only if y is solution of (2.1).*

We remind the reader that $B \partial \phi(\Lambda_E^{-1} B^* y) = \partial \psi(y)$ [26]. Problem (A.2) is related to the subdifferential inclusion:

$$f - Ay \in B \partial \phi(\Lambda_E^{-1} B^* y) \tag{A.3}$$

The relation is given by the next proposition:

PROPOSITION A.3. *y is solution of (A.3) if and only if y is solution of (A.2).*

Proof. The fact that y solution of (A.3) implies that y solution of (A.2) is a direct consequence of the definition of the subdifferential of a convex function [26]. The controverse is more complicated, and we refer the reader to chapter II.3 of [26] for a detailed proof. □

We will make use of the next lemma (whose proof is given in [8]):

LEMMA A.4. *H maximal monotone operator. Then the two following conditions are equivalent:*

(i) $u \in H(v)$

(ii) $u = H_\lambda(v + \lambda u)$

An immediate consequence of the previous lemma is the following result:

PROPOSITION A.5. *y is a solution of (A.3) if and only if (y, u) is a solution of:*

$$\begin{cases} Ay = f - Bu \\ u = H_\lambda\left(\Lambda_E^{-1} B^* y + \lambda u\right) \end{cases} \tag{A.4}$$

We are now in position to prove Theorem 2.1.

Proof. From (A.1), we get:

$$\begin{aligned} \frac{1}{\lambda^2}\left\|L_\lambda(\Lambda_E^{-1}B^*y + \lambda u) - L_\lambda(\Lambda_E^{-1}B^*y^m + \lambda u^m)\right\|_E^2 + \|u - u^{m+1}\|_E^2 \\ \leq \frac{1}{\lambda^2}\|\Lambda_E^{-1}B^*(y - y^m) + \lambda(u - u^m)\|_E^2 \\ = \|u - u^m\|_E^2 + \frac{2}{\lambda}\langle \Lambda_E^{-1}B^*(y - y^m), u - u^m\rangle_E + \frac{1}{\lambda^2}\|\Lambda_E^{-1}B^*(y - y^m)\|_E^2 \end{aligned} \tag{A.5}$$

But if we substract the first line of (2.7) to the first line of (A.4), we have: $A(y - y^m) = B(u^m - u)$. Taking the inner product with $(y - y^m)$, we deduce:

$$\langle A(y - y^m), y - y^m\rangle = \langle B(u^m - u), y - y^m\rangle = \langle u^m - u, \Lambda_E^{-1}B^*(y - y^m)\rangle \tag{A.6}$$

Hence:

$$\begin{aligned} \langle u - u^m, \Lambda_E^{-1}B^*(y - y^m)\rangle &= \langle -A(y - y^m), y - y^m\rangle \\ &\leq -\alpha\|y - y^m\|_E^2 \\ &\leq \frac{-\alpha}{\|B^*\|^2}\|\Lambda_E^{-1}B^*(y - y^m)\|_E^2 \end{aligned} \tag{A.7}$$

We now deduce from (A.5) that:

$$\begin{aligned} \frac{1}{\lambda^2}\left\|L_\lambda(\Lambda_E^{-1}B^*y + \lambda u) - L_\lambda(\Lambda_E^{-1}B^*y^m + \lambda u^m)\right\|_E^2 + \|u - u^{m+1}\|_E^2 \\ \leq \frac{1}{\lambda}\left(\frac{1}{\lambda} - \frac{2\alpha}{\|B^*\|^2}\right)\|\Lambda_E^{-1}B^*(y - y^m)\|^2 + \|u - u^m\|_E^2 \end{aligned} \tag{A.8}$$

We eventually get that, since $0 < \frac{1}{\lambda} < \frac{2\alpha}{\|B^*\|^2}$, as long as $y^m \neq y$: $\|u - u^{m+1}\|_E < \|u - u^m\|_E$. We deduce that $\|u - u^m\|_E^2$ is a convergent sequence in $\mathbb{R}$. Thus passing to the limit in (A.8), we get: $\lim_{m\to+\infty}\|\Lambda_E^{-1}B^*(y - y^m)\|_E = 0$. Using (A.7), we eventually get that $y^m \to y$.

There remains to prove that u^m also converges. We first remark that now, passing to the limit in (A.8), we get: $L_\lambda(\Lambda_E^{-1}B^*y^m + \lambda u^m) \to L_\lambda(\Lambda_E^{-1}B^*y + \lambda u)$. But since $L_\lambda = I - \lambda H_\lambda$, we get with the second line of (A.4) that: $L_\lambda(\Lambda_E^{-1}B^*y + \lambda u) = \Lambda_E^{-1}B^*y$. From the second line of (2.7), we get:

$$u^{m+1} = H_\lambda(\Lambda_E^{-1}B^*y^m + \lambda u^m) = u^m + \frac{1}{\lambda}\left(\Lambda_E^{-1}B^*y^m - L_\lambda(\Lambda_E^{-1}B^*y^m + \lambda u^m)\right) \tag{A.9}$$

Passing to the limit, we eventually get that: $\lim_{m\to+\infty}\{u^{m+1} - u^m\} = 0$. Now we can conclude that $u^m \rightharpoonup u$ in E weak, since the application

$$v \in E \to H_\lambda(\Lambda_E^{-1}B^*y(v) + \lambda v) \tag{A.10}$$

with $y(v)$ solution of: $Ay = f - Bv$, is non expansive (see [36] Corollary 4 p.199). □

Acknowledgements. The author would like to thank Vicent Caselles for fruitful discussions about Bermudez-Moreno algorithm. The author also would like to thank Antonin Chambolle for some very useful comments about a first draft of this paper. Finally, the author would like to thank Pierre Weiss for some very interesting discussions about Nesterov's schemes.

REFERENCES

[1] R. Acar and C. Vogel. Analysis of total variation penalty methods for ill-posed problems. *Inverse Problems*, 10:1217–1229, 1994.

[2] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variations and free discontinuity problems*. Oxford mathematical monographs. Oxford University Press, 2000.

[3] A. Almansa and. C. Ballester, V. Caselles, and G. Haro. A TV based restoration model with local constraints. *Journal of Scientific Computing*, 2008. To appear.

[4] F. Andreu-Vaillo, V. Caselles, and J. M. Mazon. *Parabolic quasilinear equations minimizing linear growth functionals*, volume 223 of *Progress in Mathematics*. Birkhauser, 2002.

[5] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.

[6] J.F. Aujol and G. Gilboa. Constrained and SNR-based solutions for TV-Hilbert space image denoising. *Journal of Mathematical Imaging and Vision*, 26(1-2):217–237, 2006.

[7] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle. A l1-unified variational framework for image restoration. In *ECCV 04*, volume 3024 of *Lecture Notes in Computer Sciences*, pages 1–13, 2004.

[8] A. Bermudez and C. Moreno. Duality methods for solving variational inequalities. *Comp. and Maths. with Appls.*, 7:43–58, 1981.

[9] J. Bioucas-Dias and M. Figueiredo. Thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991, 2007.

[10] X. Bresson and T. Chan. Fast minimization of the vectorial total variation norm and applications to color image processing. *UCLA CAM report*, 07-25, 2007.

[11] H. Brezis. *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*. Norht Holland, 1973.

[12] H. Brezis. *Analyse fonctionnelle. Théorie et applications.* Mathématiques appliquées pour la maitrise. Masson, 1983.

[13] A. Chambolle. An algorithm for total variation minimization and applications. *JMIV*, 20:89–97, 2004.

[14] A. Chambolle. Total variation minimization and a class of binary MRF models. In *EMMCVPR 05*, volume 3757 of *Lecture Notes in Computer Sciences*, pages 136–152, 2005.

[15] A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(3):167–188, 1997.

[16] T. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977, 1999.

[17] T. Chan and P. Mulet. On the convergence of the lagged diffusity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367, 1999.

[18] T. Chan and J. Shen. *Image processing and analysis - Variational, PDE, wavelet, and stochastic methods*. SIAM Publisher, 2005.

[19] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2), 2007.

[20] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson, 1982.

[21] P.L. Combettes and J. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9):1213–1222, 2004.

[22] P.L. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[23] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part I: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):277–291, 2006.

[24] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57:1413–1457, 2004.

[25] D. Dobson and C. Vogel. Convergence of an iterative method for total variation denoising. *SIAM Journal on Numerical Analysis*, 34:1779–1791, 1997.

[26] I. Ekeland and R. Temam. *Analyse convexe et problèmes variationnels*, volume 224 of *Grundlehren der mathematischen Wissenschaften*. Dunod, second edition, 1983.

[27] G. Facciolo, A. Almansa, J-F. Aujol, and V. Caselles. Irregular to regular sampling, denoising and deconvolution, 2008. Submitted.

[28] H. Fu, M. Ng, M. Nikolova, and J. Barlow. Efficient minimization methods of mixed l1-l1 and l2-l1 norms for image restoration. *SIAM Journal on Scientific computing*, 27(6):1881–1902, 2006.

[29] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation based image restoration. *SIAM Journal on Scientific Computing*, 27(2):622–645, 2005.

[30] Yves Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations, March 2001. The Fifteenth Dean Jacquelines B. Lewis Memorial Lectures.

[31] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers, 2004.

[32] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming (A)*, 103(1):127–152, 2005.

[33] Y. Nesterov. Gradient methods for minimizing composite objective function. *Core discussion paper*, 2007.

[34] M.K. Ng, L. Qi, Y.F. Yang, and Y. Huang. On semismooth Newton methods for total variation minimization. *Journal of Mathematical Imaging and Vision*, 27:265–276, 2007.

[35] M. Nikolova and R. Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Transactions on Image Processing*, 16(6):1623–1627, 2007.

[36] A. Pazy. On the asymptotic behavior of iterates of nonexpansive mappings in Hilbert space. *Isr. J. Math.*, 26:197–204, 1977.

[37] A. Pazy. *Semigroups of Linear Operators and Applications to Partial Differential Equations*, volume 44 of *Applied Mathematical Sciences*. Springer-Verlag, 1983.

[38] B. Polyak. *Introduction to optimization*. Optimization Software. Translation Series in Mathematics and Engineering, 2004.

[39] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[40] E. Weisfeld. Sur le point pour lequel la somme des distances de points donns est minimum. *Thoko Mathematics Journal*, 43:355–386, 1937.

[41] P. Weiss, G. Aubert, and L. Blanc-Feraud. Efficient schemes for total variation minimization under constraints in image processing. *INRIA Research Report*, 6260, 2007. http://hal.inria.fr/docs/00/16/62/30/PDF/RR-6260.pdf.