ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A fast splitting method for efficient Split Bregman iterations

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Lazzaro D., Loli Piccolomini E., Zama F. (2019). A fast splitting method for efficient Split Bregman iterations. APPLIED MATHEMATICS AND COMPUTATION, 357, 139-146 [10.1016/j.amc.2019.03.065].

Availability:
This version is available at: https://hdl.handle.net/11585/691304 since: 2019-07-09

Published:
DOI: http://doi.org/10.1016/j.amc.2019.03.065

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

12 October 2024

This is the final peer-reviewed accepted manuscript of:

Lazzaro, D., E. Loli Piccolomini, and F. Zama. "A Fast Splitting Method for Efficient Split Bregman Iterations." ***Applied Mathematics and Computation*****, vol. 357, 2019, pp. 139-146.**

The final published version is available online at : **http://dx.doi.org/10.1016/j.amc.2019.03.065**

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

A Fast Splitting Method for efficient Split Bregman Iterations ☆

D. Lazzaro[a], E. Loli Piccolomini[b], F. Zama[a]

[a] *Department of Mathematics, University of Bologna, Piazza di Porta San Donato, 5, 40126, Bologna (ITALY)*
[b] *Computer Science and Engineering Department, University of Bologna, Mura Anteo Zamboni, 7, 40126, Bologna (ITALY)*

Abstract

In this paper we propose a new fast splitting algorithm to solve the Weighted Split Bregman minimization problem in the backward step of an accelerated Forward-Backward algorithm. Beside proving the convergence of the method, numerical tests, carried out on different imaging applications, prove the accuracy and computational efficiency of the proposed algorithm.

Keywords: Weighted Total Variation, accelerated Forward Backward, FISTA, weighted Split Bregman.

1. Introduction

A large number of important image processing applications require the solution of a regularized optimization problem. In order to cope with the instrinsic ill-conditioning of the model and its sensitivity to noise, a data fit term is balanced by a weighted regularization term. Among the different regularization functions, the Total Variation (TV) and the Weighted Total Variation (WTV) have recently gained increasing attention because of their edge preserving properties [1, 2]. Therefore we focus on the numerical solution of the regularized

☆ Method for CTAN.

* Corresponding author
Email address: fabiana.zama@unibo.it (F. Zama)

minimization problem:

$$\min_u \{f(u) + \lambda\, WTV(u)\} \tag{1}$$

where $f(u)$ is the least squares fit term, $WTV(u)$ is the Weighted Total Variation regularization term and $\lambda > 0$ is the regularization parameter. The choice of the weighting function in WTV is crucial to filter out noise while preserving the image edges. In this paper we apply the non-convex *log-exp* function, proposed in [3]. The solution of problem (1) is tackled by an Accelerated Forward Backward algorithm where a modified FISTA acceleration strategy [4] is applied to the Backward step. The Weighted Split Bregman (WSB) method, used to compute the Backward step, generates a sequence of inner linear systems which constitute the computational core of the whole algorithm. In this work we propose an iterative solver (FWSB) based on a new matrix splitting which uses the matrices structure to achieve accurate and efficient solutions. Besides proving the convergence of our iterative method, we compare it to the Gauss Seidel solver on different imaging problems. The tests confirm its better performances in terms of accuracy and computational times.

The present paper is organized as follows: in section 2 we present the accelerated Forward-Backward algorithm. In section 3 the details of the Backward steps are examined and in section 4 the new splitting algorithm is introduced and its convergence proven. Finally the numerical results and conclusions are reported in sections 5, 6 respectively.

2. The Accelerated Forward Backward Algorithm

In this section we introduce the WTV function as the anisotropic discretization of the L_1 norm of the weighted gradient $(\nabla_x^w, \nabla_y^w)^T$ of the image $u \in \mathbb{R}^{N\times N}$ along the coordinate directions:

$$WTV(u) = \|\nabla_x^w u\|_1 + \|\nabla_y^w u\|_1$$

where:

$$\|\nabla_x^w u\|_1 = \sum_{i,j=1}^{N} (w_{i,j}^x)|u_x|_{i,j}, \quad \|\nabla_y^w u\|_1 = \sum_{i,j=1}^{N} (w_{i,j}^y)|u_y|_{i,j} \tag{2}$$

and $w_{i,j}^x > 0$ and $w_{i,j}^y > 0$ are constants that weight the first order differences u_x and u_y, along the vertical and horizontal directions respectively. The choice of the weights w^x and w^y is crucial when using (1) in imaging applications. In order to preserve the image edges, the weight for a pixel $u_{i,j}$ can be chosen to be inversely proportional to the local value of the gradient. Therefore the values of the weights should be small near the edges, where the gradient of the image is large, so that those components in (1) are not much penalized. Conversely, when the gradient of the image is small, like in smooth areas, the values of the weights should be large, in order to reduce variations mainly due to the presence of noise. We define the weights of the WTV, at each pixel, as the derivative of a strongly non-convex function of the gradient of the noisy image u in that pixel. In particular we choose the derivative of the non-convex *log-exp* function $\phi_\mu(t)$ as proposed in [3],

$$\phi_\mu(|t|) = \frac{1}{\log 2} \log\left(\frac{2}{1+e^{-\frac{|t|}{\mu}}}\right), \quad \mu > 0, \tag{3}$$

whose derivative is given by

$$\phi'_\mu(|t|) = \frac{1}{\mu \cdot log(2)} \frac{1}{1+e^{\frac{|t|}{\mu}}} \tag{4}$$

and satisfies:

$$\begin{cases} \phi'_\mu(|t|) \to 0 & \text{for } |t| > \mu \\ \phi'_\mu(|t|) \text{ large} & \text{for } |t| < \mu. \end{cases} \quad \mu > 0 \tag{5}$$

The function $\phi'_\mu(|\cdot|)$ approaches zero near the edges, where the gradient gets large, while it is large in smooth areas where the gradient becomes small. In fact, besides separating edges from smooth areas, $\phi'_\mu(|\cdot|)$ also identifies the small differences in intensity variations within such areas.

Since we adopt anisotropic TV discretization, our weights $w^x_{i,j}$ and $w^y_{i,j}$ are different along the x and y directions and are given by:

$$w^x_{i,j} = \phi'_\mu(|u_x|_{i,j}) = \frac{1}{\mu \cdot log(2)} \frac{1}{1 + e^{\frac{|u_x|_{i,j}}{\mu}}}, \tag{6}$$

$$w^y_{i,j} = \phi'_\mu(|u_y|_{i,j}) = \frac{1}{\mu \cdot log(2)} \frac{1}{1 + e^{\frac{|u_y|_{i,j}}{\mu}}}. \tag{7}$$

By setting the data fit function f as the least squares distance from the data z, we have

$$f(u) = \frac{1}{2}\|\Phi u - z\|_2^2 \tag{8}$$

where Φ is an $m \times n$ linear operator used to model different applications. It can be a convolution operator in the deblurring problem or a subsampling measurement operator in the compressive sensing problem, etc.

Finally we define our problem as follows:

$$\text{find} \quad u^*, \quad s.t. \quad u^* = \arg\min_u \left\{ \frac{1}{2}\|\Phi u - z\|_2^2 + \lambda \left(\|\nabla^w_x u\|_1 + \|\nabla^w_y u\|_1 \right) \right\}. \tag{9}$$

Problem (9) is convex and non differentiable and it has a unique solution under the trivial hypotheses of $\Phi \neq 0$ and u not constant.

Different methods can be used for its solution such as Chambolle Pock [5], Split-Bregman [6], Alternating Minimization [7] or the Augmented Lagrangian method **??**. All the methods should converge to the same point, with different rate. In this paper we use the Forward-Backward (FB) algorithm for the solution of the convex minimization problem (9), since it requires the tuning of very few parameters which is a great advantage in real applications.

We solve (9) by a converging sequence of Accelerated Forward-Backward steps $(v^{(n)}, u^{(n)}), n = 1, 2, \ldots$ where a modified FISTA acceleration strategy [4] is applied to the backward step. Given $u^{(0)}$, we compute for $n = 1, 2 \ldots$:

$$v^{(n)} = u^{(n-1)} + \beta \Phi^T \left(z - \Phi u^{(n-1)} \right) \tag{10}$$

$$\tilde{u}^{(n)} = \arg\min_u \left\{ \lambda \left(\|\nabla^w_x u\|_1 + \|\nabla^w_y u\|_1 \right) + \frac{1}{2\beta}\|u - v^{(n)}\|_2^2 \right\} \tag{11}$$

$$u^{(n)} = \tilde{u}^{(n-1)} + \alpha(\tilde{u}^{(n)} - \tilde{u}^{(n-1)}), \tag{12}$$

and α is chosen as follows:

$$\alpha = \frac{t_{n-1} - 1}{t_n}, \quad t_n = \frac{n + a + 1}{a}. \tag{13}$$

In our experiments, we set $a = 2$. In order to ensure the convergence of the sequence $(v^{(n)}, u^{(n)})$ to the solution of (9), the following condition on β must hold [4]:

$$0 < \beta < \frac{1}{\lambda_{max}(\Phi^T \Phi)}$$

where λ_{max} is the maximum eigenvalue in modulus. The Forward-Backward iterations are stopped with the following stopping condition:

$$\frac{\|u^{(n)} - u^{(n-1)}\|_2}{\|u^{(n)}\|_2} < \epsilon \quad \text{where} \quad \epsilon > 0. \tag{14}$$

We observe that while $v^{(n)}$ and $u^{(n)}$ are computed by explicit formulae, for the computation of $\tilde{u}^{(n)}$ in (11), we introduce a Split-Bregman strategy (section 3) and we propose a modified matrix splitting in the solution of the arising linear system.

The steps of the Accelerated Forward-Backward algorithm (AFB) are reported in algorithm 2.1.

We point out that the minimization problem (11) can be efficiently solved by means of different methods existing in literature. We cite, among others, [6] and [7]. In this paper, we use a variable splitting strategy, proposed in [2].

3. The Weighted Split Bregman Method

In this section we recall the Split Bregman method for Weighted Total Variation for the solution of (11).

Introducing two auxiliary vectors $D_x, D_y \in \mathbb{R}^{N^2}$ we rewrite (11) as a constrained minimization problem as follows:

$$\min_u \left\{ \frac{1}{2\beta} \|u - v^{(n)}\|_2^2 + \lambda \left(\|D_x\|_1 + \|D_y\|_1 \right) \right\}, \quad s.t. \;\; D_x = \nabla_x^w u, \;\; D_y = \nabla_y^w u, \tag{15}$$

Algorithm 2.1 (ALGORITHM AFB).

Input: λ, z, β, w^x, w^y ***Output:*** u^*

$u^{(0)} = \phi^T z$

$\tilde{u}^{(0)} = u^{(0)}$;

$n = 0$

`repeat`

$v^{(n+1)} = u^{(n)} + \beta\Phi^T(z - \Phi u^{(n)})$ *(Forward Step)*

Compute $\tilde{u}^{(n+1)}$ *by solving* (11) *with function FWSB in Table 4.1*

compute α *as in* (13)

$u^{(n+1)} = \tilde{u}^{(n+1)} + \alpha(\tilde{u}^{(n+1)} - \tilde{u}^{(n)})$

$n = n + 1$

`until stopping condition` *as in* (14)

$u^* = u^{(n)}$

Table 1: Accelerated Forward Backward Algorithm

where $\|D_x\|_1$ and $\|D_y\|_1$ are defined as in (2). Hence (15) can be stated in its quadratic penalized form as:

$$\min_{u,D_x,D_y} \left\{ \frac{1}{2\beta}\|u - v^{(n)}\|_2^2 + \lambda\left(\|D_x\|_1 + \|D_y\|_1\right) + \frac{\theta}{2}\left(\|D_x - \nabla_x^w u\|_2^2 + \|D_y - \nabla_y^w u\|_2^2\right) \right\} \tag{16}$$

where $\theta > 0$ represents the penalty parameter. In order to simplify the notation, exploiting the symmetry in the x and y variables, we use the subscript q indicating either x or y.

By applying the Split Bregman iterations, given an initial iterate $U^{(0)}$, we compute a sequence $U^{(1)}, U^{(2)}, \ldots, U^{(j+1)}$ by splitting (16) into three minimization problems as follows.

Given $e_x^{(0)} = e_x^{(0)} = 0$, $D_x^{(0)} = D_y^{(0)} = 0$ and $U^{(0)} = v^{(n)}$, compute:

$$U^{(j+1)} = \arg\min_u \left\{ \frac{1}{2\beta}\|u - v^{(n)}\|_2^2 + \frac{\theta}{2}\|D_x^{(j)} - \nabla_x^w u - e_x^{(j)}\|_2^2 + \frac{\theta}{2}\|D_y^{(j)} - \nabla_y^w u - e_y^{(j)}\|_2^2 \right\} \tag{17}$$

$$D_x^{(j+1)} = \arg\min_{D_x} \left\{ \lambda \|D_x\|_1 + \frac{\theta}{2} \|D_x - \nabla_x^w U^{(j+1)} - e_x^{(j)}\|_2^2 \right\}$$
$$= \mathrm{Soft}_\Lambda(\nabla_x^w U^{(j+1)} + e_x^{(j)}), \quad (18)$$

$$D_y^{(j+1)} = \arg\min_{D_y} \left\{ \lambda \|D_y\|_1 + \frac{\theta}{2} \|D_y - \nabla_y^w U^{(j+1)} - e_y^{(j)}\|_2^2 \right\}$$
$$= \mathrm{Soft}_\Lambda(\nabla_y^w U^{(j+1)} + e_y^{(j)}), \quad (19)$$

where

$$\Lambda = \frac{\lambda}{\theta} \quad (20)$$

and e_q^j is updated according to the following equation:

$$e_x^{(j+1)} = e_x^{(j)} + \nabla_x^w U^{(j+1)} - D_x^{(j+1)} = e_x^{(j)} + \nabla_x^w U^{(j+1)} -$$
$$\mathrm{Soft}_\Lambda(\nabla_x^w U^{(j+1)} + e_x^{(j)}) = \mathrm{Cut}_\Lambda(\nabla_x^w U^{(j+1)} + e_x^{(j)}), \quad (21)$$

$$e_y^{(j+1)} = e_y^{(j)} + \nabla_y^w U^{(j+1)} - D_y^{(j+1)} = e_y^{(j)} + \nabla_y^w U^{(j+1)} -$$
$$\mathrm{Soft}_\Lambda(\nabla_y^w U^{(j+1)} + e_y^{(j)}) = \mathrm{Cut}_\Lambda(\nabla_y^w U^{(j+1)} + e_y^{(j)}), \quad (22)$$

We remind that the Soft and the Cut operators apply point-wise respectively as:

$$\mathrm{Soft}_\Lambda(z) = \mathrm{sign}\,(z) \max\{|z| - \Lambda, 0\} \quad (23)$$

$$\mathrm{Cut}_\Lambda(z) = z - \mathrm{Soft}_\Lambda(z) = \begin{cases} \Lambda & z > \Lambda \\ z & -\Lambda \le z \le \Lambda \\ -\Lambda & z < -\Lambda. \end{cases} \quad (24)$$

By imposing first order optimality conditions in (17),we compute the minimum $U^{(j+1)}$ by solving the following linear system:

$$(\frac{1}{\beta} I - \theta \Delta^w) U^{(j+1)} = \frac{1}{\beta} v^{(n)} + \theta (\nabla_x^w)^T (D_x^{(j)} - e_x^{(j)}) +$$
$$\theta (\nabla_y^w)^T (D_y^{(j)} - e_y^{(j)}) \quad (25)$$

where

$$\Delta^w = -\left((\nabla_x^w)^T \nabla_x^w + (\nabla_y^w)^T \nabla_y^w \right). \quad (26)$$

Defining:

$$A = (I - \beta\theta\Delta^w) \tag{27}$$

and

$$b^{(j)} = v^{(n)} + \beta\theta(\nabla_x^w)^T(D_x^{(j)} - e_x^{(j)}) + \beta\theta(\nabla_y^w)^T(D_y^{(j)} - e_y^{(j)}) \tag{28}$$

the linear system (25) can be written as:

$$AU^{(j+1)} = b^{(j)}. \tag{29}$$

We observe that the solution of systems (29) is a crucial point because it occurs in the inner loop of the backward step, therefore it is important to employ accurate and fast methods. Since the matrix A is sparse, strictly diagonally dominant and positive definite, one of the natural choices is to use the Gauss-Seidel method.

In the next paragraph we explain the details of our proposed method named Fast Weighted Split-Bregman (FWSB) to efficiently solve (29).

4. The proposed Matrix Splitting

Exploiting the structure of the matrix A we obtain a matrix splitting of the form $E - F$ where E is the Identity matrix and F is $\beta\theta\Delta^w$. We can prove that the iterative method, based on such a splitting, is convergent if

$$0 < \theta < \frac{1}{\beta\|\Delta^w\|_\infty}. \tag{30}$$

Theorem 4.1. *Let E and F define a splitting of the matrix $A = E - F$ in* (29) *as:*

$$E = I, \quad F = \beta\theta\Delta^w. \tag{31}$$

By choosing θ as in (30) *we can prove that the spectral radius $\rho(E^{-1}F) < 1$ and, for each right-hand side B, the following iterative method*

$$X^{(m+1)} = FX^{(m)} + B, \quad m = 0, 1, \ldots \tag{32}$$

converges to the solution of the linear system $AX = B$.

In order to prove theorem 4.1 we first prove the following lemma.

Lemma 4.1. *Let $M = E + F$ where E is the Identity matrix and $F = \beta\theta\Delta^w$ with $\theta, \beta > 0$. If $0 < \theta < \frac{1}{\beta\|\Delta^w\|_\infty}$ then M is a symmetric positive definite matrix.*

Proof. By definition of Δ^w in (26), it easily follows that $M = (I + \beta\theta\Delta^w)$ is a real symmetric matrix. If we reorder lexicographically the matrix $u \in R^{N\times N}$ in a column vector of size N^2 and we use forward finite differences with zero boundary conditions, in the finite discrete setting the k-th component of the product $\Delta^\omega u$ is:

$$(\Delta^\omega u)_k = -(\alpha_k^2 + \alpha_{k+1}^2 + \eta_k^2 + \eta_{k+N}^2)u_k + \alpha_{k+1}^2 u_{k+1} + \alpha_k^2 u_{k-1} + \eta_{k+N}^2 u_{k+N} + \eta_k^2 u_{k-N} \quad (33)$$

where, from equations (6) and (7) it follows:

$$\alpha_k = \phi'_\mu(|u_x|_{i,j}) \quad \eta_k = \phi'_\mu(|u_y|_{i,j}), \quad k = (i-1)N + j \quad (34)$$

Hence on each row there are at most five non zero elements given by:

$$\begin{aligned} M_{k,k} &= 1 - \beta\theta(\alpha_k^2 + \alpha_{k+1}^2 + \eta_k^2 + \eta_{k+N}^2), \\ M_{k,k-1} &= \beta\theta\alpha_k^2, \\ M_{k,k+1} &= \beta\theta\alpha_{k+1}^2, \\ M_{k,k-N} &= \beta\theta\eta_k^2, \\ M_{k,k+N} &= \beta\theta\eta_{k+N}^2 \end{aligned} \quad (35)$$

In order to guarantee that the matrix M is positive definite, it is sufficient to determine θ such that M is strictly diagonally dominant, namely

$$\left|1 - \beta\theta(\alpha_k^2 + \alpha_{k+1}^2 + \eta_k^2 + \eta_{k+N}^2)\right| > \beta\theta(\alpha_k^2 + \alpha_{k+1}^2 + \eta_k^2 + \eta_{k+N}^2) \quad \forall\ k = 1, ..., N^2.$$

Hence:

$$\frac{1}{\beta\theta} > 2(\alpha_k^2 + \alpha_{k+1}^2 + \eta_k^2 + \eta_{k+N}^2) \quad \forall\ k = 1, ..., N^2$$

It easily follows that this relation is satisfied for

$$\beta\theta < \max_{k=1,..N^2} \left(2(\alpha_k^2 + \alpha_{k+1}^2 + \eta_k^2 + \eta_{k+N}^2)\right)$$

namely,

$$0 < \theta < \frac{1}{\beta\|\Delta^w\|_\infty} \tag{36}$$

□

Proof of theorem 4.1

Proof. Using the Householder-Johns theorem [9, 10] $\rho(E^{-1}F) < 1$ iff $A = E - F$ is symmetric positive definite (SPD) and $E^* + F$ is symmetric and positive definite, where E^* is the conjugate transpose of E. The matrix A is SPD since it is symmetric and strictly diagonal dominant. From Lemma 4.1, we have $E^* + F = M$ and therefore the condition on θ guarantees that $E^* + F$ is symmetric and positive definite. □

Hence we compute the Weighted Split Bregman solution $U^{(j+1)}, j = 0, 1, \ldots$ by means of the iterative method defined in (32) with $B = b^{(j)}$ as in (28).

Substituting (26) in (32) we have:

$$X^{(m+1)} = -\beta\theta \left((\nabla_x^w)^T \nabla_x^w + (\nabla_y^w)^T \nabla_y^w\right) X^{(m)} + b^{(j)}. \tag{37}$$

By substituting (28) in (37) and collecting ∇_x^w, ∇_y^w we obtain :

$$X^{(m+1)} = v^{(n)} + \beta\theta \Big[(\nabla_x^w)^T(-\nabla_x^w X^{(m)} + D_x^{(j)} - e_x^{(j)}) + \\ +(\nabla_y^w)^T(-\nabla_y^w X^{(m)} + D_y^{(j)} - e_y^{(j)})\Big]. \tag{38}$$

In Table 4.1 we report the function Fast Weighted Split Bregman (FWSB) for the solution of problem (11). The output variable $U^{(j)}$ is the computed solution and $\bar{m}$ is the number of total iterations.

The stopping condition of both the loops (with indices j and m) is defined on the basis of the relative tolerance parameter τ as follows:

$$\|w^{(k+1)} - w^{(k)}\| \leq \tau \|w^{(k))}\| \tag{39}$$

Algorithm 4.1.

*[$U^{(j)}$, $\bar{m}$]=**FWSB(**$\lambda, \theta, \beta, v^{(n)}, w^x, w^y$**)***

$\Lambda = \frac{\lambda}{\theta}$, $\bar{m} = 0$,

$U^{(0)} = v^{(n)}$, $e_x^{(0)} = e_y^{(0)} = 0$;

$j = 1$

repeat

$U_x = \nabla_x^w U^{(j-1)}$; $U_y = \nabla_y^w U^{(j-1)}$;

$e_x^{(j)} = Cut_\Lambda(z_x)$; $e_y^{(j)} = Cut_\Lambda(z_y)$;

$m = 0$

repeat

$X^{(m+1)}$ *computed as in* (38)

$m = m + 1$

until `stopping condition` (39)

$U^{(j+1)} = X^{(m)}$;

$\bar{m} = \bar{m} + m; j = j + 1$

until `stopping condition` (39)

Table 2: FWSB Algorithm for the solution of problem (11)

where $w^{(k)} \equiv U^{(j)}$ in the outer loop ($k \equiv j$) and $w^{(k)} \equiv X^{(m)}$ in the inner loop ($k \equiv m$).

5. Numerical Experiments

In this section we analyze the results obtained by applying Algorithm 2.1 to two representative test problems related to different image processing applications. The minimization problem (9) is solved applying the accelerated Forward-Backward method together with the Weighted Split Bregman method. Our aim is to compare the proposed FWSB method with Gauss-Seidel (WSB_GS) applied to the linear system (27). The experiments are performed on a PC intel i7 with 32 Gbyte Ram, by using Matlab R2018a.

In test problem T1, we consider an image deblurring problem where the matrix Φ in (9) is a Gaussian blur operator obtained by the Matlab function `fspecial`, with standard deviation $\sigma = 1.5$ and size 9. The algorithms are tested both on noiseless and noisy data. The results reported here are relative to the case of Gaussian white noise with variance $\delta = 0.5 \cdot 10^{-2}$.

Test problem T2 is a compressed sensing application, where z represents subsampled Magnetic Resonance data in the so called Kspace and the matrix Φ in (8) is the undersampled Fourier matrix, obtained by the Hadamard product between the full resolution Fourier matrix $\mathbf{F}$ and the mask $\mathcal{M}$, i.e.

$$\Phi = \mathcal{M} \circ \mathbf{F}. \tag{40}$$

In the tests reported in the present work we consider $\mathcal{M}$ as a radial mask with sampling percentages $S_p = 3.98\%$ and $S_p = 4.3\%$ relative to 8 and 10 radial lines respectively.

The quality of the reconstructed image is evaluated by means of the Peak Signal to Noise Ratio (PSNR)

$$PSNR = 20 \log_{10} \frac{\max(x)}{rmse}, \text{ where } rmse = \sqrt{\frac{\sum_i \sum_j (u_{i,j} - x_{i,j})^2}{N^2}}.$$

where x is the reference true image and u is the reconstructed image. In the tests reported in this section x is the Shepp Logan Phantom (256×256 pixels).

Since our purpose is to evaluate the best possible solution obtained by each method, in all tests the regularization parameter λ is heuristically set to the best possible value with respect to PSNR.

Test	Noise	FWSB			WSB_GS		
		PSNR	a_it	time	PSNR	a_it	time
T1	$\delta = 0$	25.5	4	$7.20s$	25.27	145	$50.64s$
	$\delta = 5 \cdot 10^{-3}$	24.38	5	$5.74s$	24.29	142	$15.17s$
T2 ($S_p = 4.30$)	$\delta = 0$	36.22	15	$3.36s$	33.32	135	$12.64s$
	$\delta = 5 \cdot 10^{-3}$	35.86	15	$4.58s$	32.93	135	$14.18s$
T2 ($S_p = 3.98$)	$\delta = 0$	28.91	15	$5.25s$	27.86	138	$17.94s$
	$\delta = 5 \cdot 10^{-3}$	28.66	15	$5.12s$	27.45	138	$18.10s$

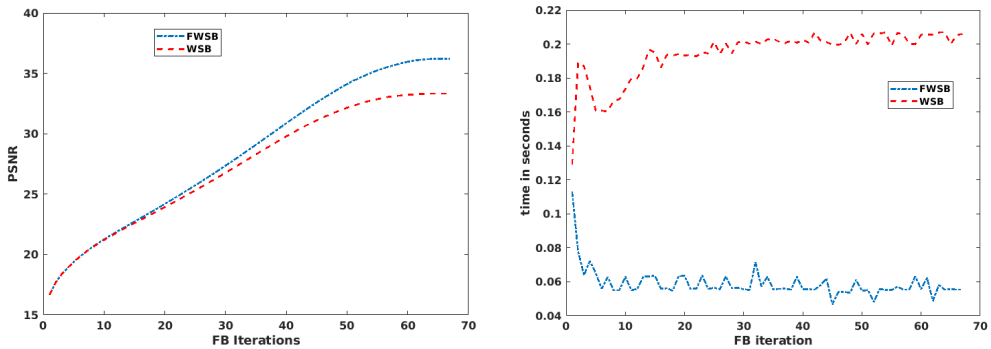
Table 3: Values of PSNR, avearge iteration number a_it and times for the different tests. The noise variance δ is reported for each test.

In table 3 we report the PSNR (columns 3, 6), the computation times (columns 5, 8) and the average number of linear solver iterations (columns 4, 7), obtained by averaging the total number of linear solver iterations over the number of forward backward steps.

We observe that FWSB is always the most efficient method since it obtaines in each case the smallest average number of iterations and the shortest computation time. Regarding the accuracy, we can see that FWSB always reaches the greatest values of PSNR. In figures 1 and 2 we can appreciate the evolution of PSNR and computation times after each FB iteration for both FWSB and WSB_GS and we remark the better performance of FWSB in terms of accuracy and computation times.

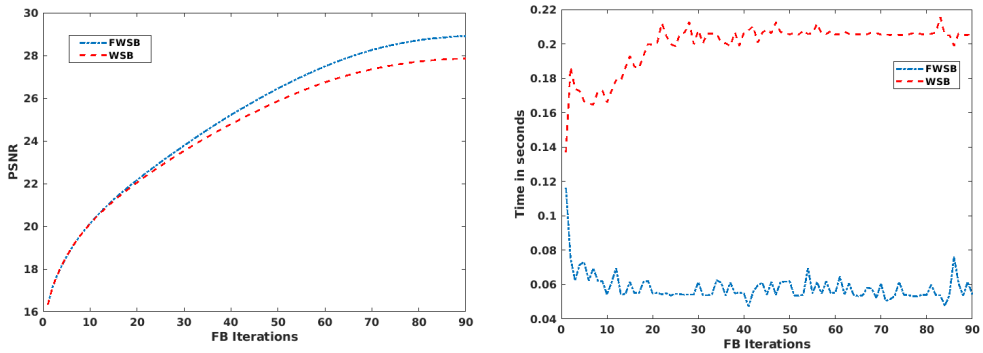
6. Conclusions

In this work we proposed a fast splitting method FWSB for the solution of the inner step of the Weighted Split Bregman method. We proved its convergence and compared it to the most commonly used iterative methods i.e.



(a) (b)

Figure 1: T2 test $S_p = 4.30$, FWSB, blue dash-dot line; WSB_GS, red dashed line. (a) PSNR vs. FB iterations. (b) Time in seconds(s.) vs FB iterations.



(a) (b)

Figure 2: T2 test $S_p = 3.98$: FWSB, blue dash-dot line; WSB_GS, red dashed line. (a) PSNR vs. FB iterations. (b) Time in seconds(s.) vs FB iterations.

Gauss-Seidel. After running a large set of experiments for different problems and datasets, we reported the most representative results obtained in the case of image deblurring and sparse MRI. From the results we can state that FWSB is a very efficient and accurate method to be used in the solution of Weighted Split Bregman iterations.

Acknowledgements

This research has been partially supported by GNCS-INDAM Italy.

References

[1] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D 60 (1992) 259–268.

[2] X. Zhang, M. Burger, X. Bresson, S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM J. IMAGING SCIENCES 3 (2010) 253–276.

[3] L. B. Montefusco, D. Lazzaro, S. Papi, A fast algorithm for nonconvex approaches to sparse recovery problems, Signal Processing 93 (9) (2013) 2636 – 2647.

[4] A. Chambolle, C. Dossal, On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm ", Journal of Optimization Theory and Applications 166 (3) (2015) 968–982. doi:10.1007/s10957-015-0746-4

[5] A. Chambolle, T. Pock, A first order primal-dual algorithms for convex problems with applications to imaging, J. Math. Imag. vision 40 (2010) 120–145.

[6] T. Goldstein, S. Osher, The split bregman method for l1-regularized problems, SIAM Journal on Imaging Sciences 2 (2) (2009) 323–343.

[7] Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, SIAM Journal on Imaging Sciences 1 (3) (2008) 248–272.

[8] C. Wu, X. Tai, Augmented Lagrangian Method, Dual Methods, and Split Bregman Iteration for ROF, Vectorial TV, and High Order Models, SIAM Journal on Imaging Sciences 3 (3) (2010) 300-339.

[9] J. Ortega, Matrix Theory: a Second Course, pub-PLENUM, 1987.

[10] Z.-H. Cao, A note on p-regular splitting of hermitian matrix, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1392–1393.