



Proximal Algorithm Meets a Conjugate descent

Matthieu Kowalski

► **To cite this version:**

Matthieu Kowalski. Proximal Algorithm Meets a Conjugate descent. Pacific journal of optimization, Yokohama Publishers, 2016, 12 (3), pp.669-695. hal-00505733v3

HAL Id: hal-00505733

<https://hal.archives-ouvertes.fr/hal-00505733v3>

Submitted on 26 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proximal Algorithm Meets a Conjugate Descent

Matthieu Kowalski *

September 26, 2011

Abstract

This paper proposes an enhancement of the non linear conjugate gradient algorithm for some non-smooth problems. We first extend some results of descent algorithms in the smooth case for convex non-smooth functions. We then construct a conjugate descent algorithm based on the proximity operator to obtain a descent direction. We finally provide a convergence analysis of this algorithm, even when the proximity operator must be computed by an iterative process. Numerical experiments show that this kind of method has some potential, even if proposed algorithms do not outperform accelerated first order algorithm yet.

1 Introduction

A common and convenient formulation when dealing with an inverse problem is to model it as a variational problem, giving rise to a convex optimization problem. In this article, we focus on the following formulation:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} F(x) = f_1(x) + f_2(x), \quad (1)$$

assuming that

Assumption 1.

- f_1 is a proper convex lower semi-continuous function, L -Lipschitz differentiable, with $L > 0$,
- f_2 is a non-smooth proper convex lower semi-continuous function,
- F is coercive finite function with $\text{dom}(F) = \mathbb{R}^N$

*Laboratoire des Signaux et Systèmes, UMR 8506 CNRS - SUPELEC - Univ Paris-Sud 11, 91192 Gif-sur-Yvette Cedex, France Tel.: +33 (0)1 69 85 17 47 Fax.: +33 (0)1 69 85 17 65 kowalski@lss.supelec.fr

A wide range of inverse problems belongs to this category. In the past decades, several algorithms have been proposed to deal with this general framework, intensively used in the signal processing community, as stressed in Combettes *et al.* [9]. An outstanding illustration concerns regularized or constrained least squares. For about 15 years, the convex non-smooth $\ell_2 - \ell_1$ case, known as Basis Pursuit (Denoising) [8] in signal processing or as Lasso [29] in machine learning and statistics, has been widely studied both in a theoretical and practical point of view. This specific problem highlights interesting properties, in particular the sparsity principle which finds a typical application in the compressive sensing [11],[7].

Within the general framework given by (1) and Assumption 1,¹ we aim to generalize a classical algorithm used in smooth optimization: the non-linear conjugate gradient algorithm. To solve Problem (1), we propose to take advantage of the forward-backward proximal approach to find a good descent direction and to construct a practical conjugate descent algorithm. To our knowledge, such a method has not been proposed in this context, although a generalization of the steepest residual methods was proposed in the past for non-smooth problems [32].

The paper is organized as follows. Section 2 recalls definitions and results on convex analysis. In Section 3, we give a brief state of the art concerning the methods that deal with Problem (1), and describe more precisely the two algorithms which inspired ours: the forward-backward proximal algorithm [9] and the non-linear conjugate gradient method [26]. We then extend some results known in the smooth case for (conjugate) gradient descent to the non-smooth case in Section 4. Hence, we derive and analyze the resulting algorithm in Section 5. Finally, Section 6 presents some numerical illustrations.

2 Reminder on convex analysis

This section is devoted to important definitions, properties and theorems issued from convex analysis, which is intensively used in the rest of the paper. First, we focus on directional derivatives and subgradients which are important concepts to deal with non differentiable functionals. In this context, we define what we call a *descent direction* and give some important properties used to establish results of convergence in the following sections. Finally, the foundations concerning proximity operators are recalled together with an important theorem of convex optimization.

Definition 1 (Directional derivative). *Let F be a lower semi-continuous convex function on \mathbb{R}^N . Then, for all $x \in \mathbb{R}^N$, for all $d \in \mathbb{R}^N$, the directional derivative exists and is defined by*

$$F'(x; d) = \lim_{\lambda \downarrow 0} \frac{F(x + \lambda d) - F(x)}{\lambda} .$$

¹In here and what follows, the denomination Problem (1) refers to this combination.

We also give the definition of the subdifferential which is a significant notion of convex analysis.

Definition 2 (Subdifferential). *Let F be a lower semi-continuous convex function on \mathbb{R}^N . The subdifferential of F at x is the subset of \mathbb{R}^N defined by*

$$\partial F(x) = \{g \in \mathbb{R}^N, F(y) - F(x) \geq \langle g, y - x \rangle \text{ for all } y \in \mathbb{R}^N\},$$

or equivalently

$$\partial F(x) = \{g \in \mathbb{R}^N, \langle g, d \rangle \leq F'(x; d) \text{ for all } d \in \mathbb{R}^N\}.$$

An element of the subdifferential is called a subgradient. A consequence of this definition is that

$$\sup_{g \in \partial F(x)} \langle g, d \rangle = F'(x; d),$$

and we denote by

$$g_s(x; d) = \arg \sup_{g \in \partial F(x)} \langle g, d \rangle. \quad (2)$$

As we are interested in descent methods for optimization, we recall the definition of a descent direction as in [16].

Definition 3 (Descent direction). *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function. d is a descent direction for F at x if and only if there exists $\alpha > 0$ such that $F(x + \alpha d) \leq F(x)$.*

A direct consequence of this definition, is that such a direction exists if and only if x is not a minimum of F . More precisely, we have the following proposition useful for convex optimization.

Proposition 1. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function. d is a descent direction for F at x if and only if, for all $g \in \partial F(x)$, $\langle d, g \rangle \leq 0$.*

If the large inequalities in Definition 3 and Proposition 1 are replaced by strict inequalities, then d is called a strict descent direction.

In order to prove some convergence results we also need the following proposition, that specify some kind of continuity properties of the subgradient (one can refer to [5, sec. 8.2.2, p. 106])

Proposition 2. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function and $\partial F(x)$ its subdifferential at x . Then the operator $x \mapsto \partial F(x)$ has a closed graph. i.e, for any sequences $\{x_k\}$ of \mathbb{R}^N such that $\lim_{k \rightarrow \infty} x_k = \bar{x}$, and $g_k \in \partial F(x_k)$ such that $\lim_{k \rightarrow \infty} g_k = \bar{g}$, then*

$$\bar{g} \in \partial F(\bar{x}).$$

However, as stressed in [5], we *do not have* in general:

$$x_k \rightarrow \bar{x}, \bar{g} \in \partial F(\bar{x}) \Rightarrow \exists g_k \in \partial F(x_k) \rightarrow \bar{g}.$$

Because of this lack of continuity, the steepest descent method for non-smooth convex functions does not necessarily converge (see [5] for a counter example).

As this work is based on the forward-backward algorithm, we also deal with the proximity operator introduced by Moreau [19], which is intensively used in convex optimization algorithms.

Definition 4 (Proximity operator). *Let $\varphi : \mathbb{R}^N \rightarrow \mathbb{R}$ be a lower semi-continuous convex function. The proximity operator associated with φ denoted by $\text{prox}_\varphi : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is given by*

$$\text{prox}_\varphi(y) = \frac{1}{2} \arg \min_{x \in \mathbb{R}^N} \{ \|y - x\|_2^2 + \varphi(x) \} . \quad (3)$$

Furthermore, proximity operators are firmly non expansive, hence continuous (See [9] for more details concerning proximity operators).

To conclude this section, we state an important theorem of convex optimization [25], usefull to prove convergence of optimization algorithm in a finite dimensional setting.

Theorem 1. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function, which admits a set of minimizer X^* . Let $\{x_k\}$ be a sequence satisfying $\lim_{k \rightarrow \infty} F(x_k) = F(x^*)$, with $x^* \in X^*$. Then all convergent subsequences of $\{x_k\}$ converge to a point of X^* .*

Before going further into the proximal-conjugate algorithm, we present a brief state of the art of the main existing algorithms in convex optimization. A particular attention is paid on the two algorithms which inspire the present paper.

3 State of the art

We first expose the non-linear conjugate gradient algorithm for smooth functions, and then the Iterative Shrinkage/Thresholding Algorithm (ISTA). We conclude by a short review of popular algorithms used for convex non-smooth optimization.

3.1 Non-linear conjugate gradient (NLCG)

The conjugate gradient algorithm was first introduced to minimize quadratic functions [15], and was extended to minimize general smooth functions (non necessarily convex). This extension is usually called the non-linear conjugate gradient algorithm. There exists an extensive literature about the (non-linear) conjugate gradient. One can refer to the popular paper of Shewchuck [28] available on line, but also to the book [26] of Pytlak dedicated to conjugate gradient algorithms or to the recent survey [14].

The non-linear conjugate gradient algorithm has the following form:

Algorithm 1 (NLCG). *Initialization: Choose $x_0 \in \mathbb{R}^N$. Repeat until convergence:*

1. $p_k = -\nabla F(x_k)$
2. $d_k = p_k + \beta_k d_{k-1}$
3. choose a step length $\alpha_k > 0$
4. $x_{k+1} = x_k + \alpha_k d_k$

where β_k is the conjugate gradient update parameter that belongs to \mathbb{R} . Various choices can be made for β_k . Some of the most popular are

$$\beta_k^{HS} = \frac{\langle \nabla F(x_{k+1}), \nabla F(x_{k+1}) - \nabla F(x_k) \rangle}{\langle d_k, \nabla F(x_{k+1}) - \nabla F(x_k) \rangle}, \quad (4)$$

$$\beta_k^{FR} = \frac{\|\nabla F(x_{k+1})\|^2}{\|\nabla F(x_k)\|^2}, \quad (5)$$

$$\beta_k^{PRP} = \frac{\langle \nabla F(x_{k+1}), \nabla F(x_{k+1}) - \nabla F(x_k) \rangle}{\|\nabla F(x_k)\|^2}. \quad (6)$$

β_k^{HS} was proposed in the original paper of Hestenes and Stiefel [15]; β_k^{FR} , introduced by Fletcher and Reeves [13], is useful for some results of convergence as in [1]; β_k^{PRP} , by Polak and Ribière [23] and Polyak [24], is known to have good practical behavior. One can refer to [14] for a more exhaustive presentation of the possible choices for β_k .

3.2 Forward-backward proximal algorithm

A simple algorithm used to deal with functionals as (1) is ISTA, also known as Thresholded Landweber [10] or forward-backward proximal algorithm [9]. Let us recall that f_1 must be L -Lipschitz differentiable.

Algorithm 2 (ISTA). *Initialization:* choose $x_0 \in \mathbb{R}^N$.
Repeat until convergence:

1. $x_{k+1} = \text{prox}_{\mu f_2}(x_k - \mu \nabla f_1(x))$

where $0 < \mu < 2/L$.

Remark 1. *Computation of the prox.*

As one of the aims of this contribution is to connect conjugate descents methods and the proximal method, let us rewrite Algorithm 2 as a descent algorithm with a constant step size equals to one. First, we give in Algorithm 3 the general form of a descent algorithm.

Algorithm 3 (General descent algorithm). *Initialization:* choose $x_0 \in \mathbb{R}^N$.
Repeat until convergence:

1. choose a descent direction d_k
2. choose a step length $\alpha_k > 0$

$$3. x_{k+1} = x_k + \alpha_k d_k$$

Then, we can prove that $s_k = \text{prox}_{\mu f_2}(x_k - \mu \nabla f_1(x)) - x_k$ is a descent direction with $\mu < \frac{2}{L}$. Indeed, since f_1 is convex L -Lipschitz differentiable,

$$0 \leq f_1(x) - f_1(y) - \langle \nabla f_1(y), x - y \rangle \leq L/2 \|x - y\|^2. \quad (7)$$

Hence, by introducing the surrogate

$$F^{sur}(x, y) = f_1(y) + \langle \nabla f_1(y), x - y \rangle + \frac{1}{\mu} \|x - y\|^2 + f_2(x), \quad 0 < \mu < \frac{2}{L} \quad (8)$$

we have for all $x, y \in \mathbb{R}^N$

$$F(x) = F^{sur}(x, x) \leq F^{sur}(x, y). \quad (9)$$

Let us denote by x_{k+1} the minimizer of $F^{sur}(\cdot, x_k)$. Then, one has [31, p. 30]

$$\begin{aligned} x_{k+1} &= \arg \min_x F^{sur}(x, x_k) \\ &= \arg \min_x \frac{1}{2\mu} \|x - x_k + \mu \nabla f_1(x_k)\|^2 - \frac{1}{2\mu} \|\mu \nabla f_1(x_k)\|^2 + f_2(x) \\ &= \arg \min_x \frac{1}{2} \|x - x_k + \mu \nabla f_1(x_k)\| + \mu f_2(x) \\ &= \text{prox}_{\mu f_2}(x_k - \mu \nabla f_1(x_k)). \end{aligned}$$

Such a choice assures to decrease the value of the functional:

$$\begin{aligned} f_1(x_{k+1}) + f_2(x_{k+1}) &= F^{sur}(x_{k+1}, x_{k+1}) \\ &\leq F^{sur}(x_{k+1}, x_k) \\ &\leq F^{sur}(x_k, x_k) \\ &\leq f_1(x_k) + f_2(x_k). \end{aligned}$$

Consequently, $s_k = x_{k+1} - x_k$ is a descent direction for F at x_k , and we can write algorithm 2 as a descent algorithm with a constant step size $\alpha_k = 1$ for all k :

Algorithm 4 (ISTA as a descent algorithm). *Initialization: choose $x_0 \in \mathbb{R}^N$. Repeat until convergence:*

1. $p_k = \text{prox}_{\mu f_2}(x_k - \mu \nabla f_1(x))$
2. $s_k = p_k - x_k$
3. $x_{k+1} = x_k + s_k$

It is well known that ISTA converges to a minimizer of F (see [9], [10]). We can state the following corollary of this convergence results.

Corollary 1. *Let F be the function defined in (1). Let $\{x_k\}$ be generated by the descent algorithm 3, and let $p_k = \text{prox}_{\mu f_2}(x_k - \frac{1}{L}\nabla f_1(x_k))$, with $0 < \mu < 2/L$. If $\lim_{k \rightarrow \infty} x_k - p_k = 0$, then all convergent subsequences of $\{x_k\}$ converge to a minimizer of F .*

Proof. $F(x_k)$ is a decreasing sequence bounded from below. As F is continuous and stand in a finite dimensional space, one can extract a convergent subsequence of $\{x_k\}$, denoted by $\{\tilde{x}_k\}$, with \tilde{x} being its limit. As the proximity operator is continuous, let $\{\tilde{p}_k\}$ being the corresponding subsequence of $\{p_k\}$ obtained from $\{x_k\}$.

Then, for $\varepsilon > 0$, there exists $K > 0$ such that for all $k > K$, we have (by hypothesis) $\|\tilde{p}_k - \tilde{x}_k\| < \varepsilon/2$ and $\|\tilde{x}_k - \tilde{x}\| < \varepsilon/2$. Hence, for all $k > K$, $\|\tilde{p}_k - \tilde{x}\| \leq \|\tilde{p}_k - \tilde{x}_k\| + \|\tilde{x}_k - \tilde{x}\| < \varepsilon$. Thus, \tilde{x} is proven to be a fixed point of $\text{prox}_{\frac{1}{L}f_2}(\cdot - \frac{1}{L}\nabla f_1(\cdot))$. Moreover, one can state that \tilde{x} is a minimizer of F , using Proposition 3.1 from [9].

Finally, Theorem 1 leads to Corollary 1. ■

3.3 Other algorithms

As already mentioned in the introduction, various range of algorithms were developed during the last past years. In particular, one can cite algorithms inspired by the significant works of Nesterov [21, 20], such as the Beck and Teboulles's Fast Iterative Shrinkage/Thresholding Algorithm (FISTA) [3]. The main advantages of these algorithms is the speed of convergence, in $\mathcal{O}(\frac{1}{k^2})$, where k is the number of iterations, which must be compared to the speed of ISTA in $\mathcal{O}(\frac{1}{k})$. This theoretical results are often verified in practice: ISTA is much slower than FISTA to reach a good estimation of the sought minimizer. In [30], Paul Tseng gives a good overview, with generalizations and extensions of such accelerated first order algorithm. Other accelerated algorithms were proposed, such as SPARSA by Wright *et al.* [33] or the alternating direction methods via the augmented Lagrangian [22].

4 A general conjugate descent algorithm

In this section, we generalize some theoretical results known for gradient descent in the smooth case, to a general descent algorithm which can be used to minimize a convex, non smooth, functional. We first present a general conjugate descent algorithm, not studied yet in the non smooth case, and discuss the choice of the step length thanks to an extension of the Wolfe conditions defined in the smooth case (see for example [4, 26]). We then study the convergence of the algorithm for different choices of the step length. For this purpose, we extend the notion of "uniformly gradient related" descent proposed by Bertsekas [4] and generalize Al-Baali's result [1], which assures that the conjugation provides a descent direction under some conditions for the choice of the conjugate parameter.

4.1 A general (conjugate) descent algorithm for non-smooth functions

We extend the non linear conjugate gradient Algorithm 1 by presenting the following general conjugate descent algorithm.

Algorithm 5. *Initialization: choose $x_0 \in \mathbb{R}^N$.
Repeat until convergence:*

1. find s_k , a descent direction at x_k for F
2. choose β_k , the conjugate parameter
3. $d_k = s_k + \beta_k d_{k-1}$
4. find a step length $\alpha_k > 0$
5. $x_{k+1} = x_k + \alpha_k d_k$

When $\beta_k = 0$ this algorithm obviously reduces to a classical general descent algorithm as Algorithm 3 with an adaptive step length. The choice of β_k will be discussed later in the paper (see Theorem 3).

Ideally, one would find the optimal step size α_k . However, in the general case, one does not have access to a closed form of this quantity, then a line search must be performed.

4.2 (Modified) Wolfe conditions

Wolfe conditions are usually defined for smooth functions in order to perform a line search of a proper step size. These conditions were extended to convex, not necessarily differentiable, functions in [34]. At each iteration k , let x_k be updated as in step 5 of Algorithm 5. One can perform a line search to choose the step size α_k in order to verify the Wolfe conditions which are:

$$F(x_k + \alpha_k d_k) - F(x_k) \leq c_1 \alpha_k \langle g_s(x_k; d_k), d_k \rangle \quad (10)$$

$$\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle \geq c_2 \langle g_s(x_k; d_k), d_k \rangle, \quad (11)$$

with $0 < c_1 < c_2 < 1$, and g_s defined in (2).

As in the smooth case, one can extend these conditions to obtain the strong Wolfe conditions by replacing (11) by

$$|\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle| \leq -c_2 \langle g_s(x_k; d_k), d_k \rangle. \quad (12)$$

In [34], the authors prove that such a step size α_k exists. For non smooth problems, Mifflin proposed in [18] other conditions:

$$F(x_k + \alpha_k d_k) - F(x_k) \leq -c_1 \alpha_k \|d_k\|^2 \quad (13)$$

$$\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle \geq -c_2 \|d_k\|^2, \quad (14)$$

with $0 < c_1 < c_2 < 1$. We will refer to these conditions as the Mifflin-Wolfe conditions in the following. Mifflin proposed also a procedure which converges in a *finite number of iterations* to a solution α satisfying the Mifflin-Wolfe conditions. The procedure is the following:

Algorithm 6 (Line search). *Initialization: Choose $\alpha > 0$. Set $\alpha_L = 0, \alpha_N = +\infty$.*

Repeat until α verifies (13) and (14)

1. *If α verifies (13) set $\alpha_L = \alpha$
Else $\alpha_N = \alpha$*
2. *If $\alpha_N = +\infty$ set $\alpha = 2\alpha$
Else $\alpha = \frac{\alpha_L + \alpha_N}{2}$*

Now that we have defined rules to choose the step length, we pay attention to the convergence properties of Algorithm 5.

4.3 Convergence results

In order to state some results on the convergence of Algorithm 5, we adapt Proposition 1.8 of [4] in the non differentiable case. For that, we first adapt the definition of the *uniformly gradient related descent* of [4]:

Definition 5. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function, and $\partial F(x)$ its subdifferential at x . Let $\{x_k\}$ be a sequence generated by a descent method, with $x_{k+1} = x_k + \alpha_k d_k$. The sequence $\{d_k\}$ is uniformly subgradient related to $\{x_k\}$ if for every convergent subsequence $\{x_k\}_{k \in K}$ for which*

$$0 \notin \lim_{k \rightarrow +\infty, k \in K} \partial F(x_k),$$

there holds

$$0 < \liminf_{k \rightarrow +\infty, k \in K} |F'(x_k; d_k)|, \quad \limsup_{k \rightarrow +\infty, k \in K} |d_k| < +\infty.$$

We can now state the following theorem.

Theorem 2. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function. Assume that $\{x_k\}$, $\{d_k\}$ and α_k are the sequences generated by Algorithm 5. Assume that for all k , d_k is a uniformly subgradient related descent direction. If α is a choosen :*

- *to be a constant step size;*
- *or, to satisfy the Mifflin-Wolfe conditions;*
- *or, to be the optimal step size;*
- *or to satisfy the Wolfe conditions,*

then every convergent subsequences of x_k converge to a minimum of F .

Proof. We provide here the proof for the Mifflin-Wolfe conditions. The proof in the other cases is straightforward. Since d_k is a descent direction, the sequence of $F(x_k)$ is decreasing, and as it is bounded from below, converges to some F^* .

Then $\sum_{k=0}^{+\infty} F(x_k) - F(x_{k+1}) < +\infty$.

From the first Mifflin-Wolfe condition, we can state that

$$\lim_{k \rightarrow +\infty} \alpha_k \|d_k\|^2 = 0 .$$

Let $\{\tilde{x}_k\} = \{x_k\}_{k \in \mathcal{K}}$ be a convergent subsequence of $\{x_k\}$ converging to \tilde{x} , and suppose that \tilde{x} is not a minimum of F . Since $\{d_k\}$ is uniformly subgradient related, we have $0 < \liminf_{k \rightarrow +\infty, k \in \mathcal{K}} |F'(x_k; d_k)|$ and then $\lim_{k \rightarrow +\infty, k \in \mathcal{K}} \alpha_k = 0$.

During Algorithm 6, we can thus find α such that:

$$F(x_k + \alpha d_k) - F(x_k) > -c_1 \alpha \|d_k\|^2 .$$

Thus,

$$F(x_k + \alpha d_k) - F(x_k + \alpha_k d_k) > -c_1 (\alpha - \alpha_k) \|d_k\|^2 ,$$

and because F is convex we have (see [18])

$$\liminf_{\alpha \downarrow \alpha_k} \langle g_s(x_k + \alpha d_k; d_k), d_k \rangle \geq \limsup_{\alpha \downarrow \alpha_k} \frac{F(x_k + \alpha d_k) - F(x_k + \alpha_k d_k)}{\alpha - \alpha_k} \geq -c_1 \|d_k\|^2 .$$

Thanks to proposition 2, there exists $K > 0$, such that for all $k > K$, $k \in \mathcal{K}$ we have

$$\langle g_s(x_k + \alpha_k d_k; d_k), d_k \rangle \leq \langle g_s(x_k; d_k), d_k \rangle . \quad (15)$$

Therefore,

$$\langle g_s(x_k; d_k), d_k \rangle \geq -c_1 \|d_k\|^2 ,$$

i.e.

$$c_1 \geq \frac{|\langle g_s(x_k; d_k), d_k \rangle|}{\|d_k\|^2} .$$

Then, as $c_1 < 1$, for $k > K$ $k \in \mathcal{K}$ we have

$$|\langle g_s(x_k; d_k), d_k \rangle| \leq \|d_k\|^2 .$$

From the second Mifflin-Wolfe condition, we obtain that for all $k > K$, $k \in \mathcal{K}$:

$$\begin{aligned} \langle g_s(x_{k+1}; d_k) - g_s(x_k; d_k), d_k \rangle &= \langle g_s(x_{k+1}; d_k), d_k \rangle - \langle g_s(x_k; d_k), d_k \rangle \\ &\geq -c_2 \|d_k\|^2 - \langle g_s(x_k; d_k), d_k \rangle \\ &\geq (1 - c_2) \|d_k\|^2 , \end{aligned}$$

with $c_2 < 1$, contradicting (15). Then \tilde{x} is a minimum of F . ■

4.4 A uniformly subgradient related conjugation

In the case of an optimal choice of the step size, we are sure that at each iterations, d_k is a descent direction.

Lemma 1. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function under Assumption 1. Let $\alpha_k^* = \arg \min_{\alpha > 0} F(x_k + \alpha d_k)$, where d_k is a descent direction for F at x_k . If s_{k+1} is a descent direction for F at $x_{k+1} = x_k + \alpha_k^* d_k$, then for all $\beta_k > 0$, $d_{k+1} = s_{k+1} + \beta_k d_k$ is a descent direction for F at x_{k+1} .*

Moreover, if s_k is uniformly subgradient related and if, $\lim_{k \rightarrow +\infty} |\beta_k| < 1$, then d_k is uniformly subgradient related.

Proof. As F is a finite convex function on \mathbb{R}^N , we can apply [16, Theorem 4.2.1] which leads to

$$\partial_\alpha F(x_k + \alpha d_k) = \langle d_k, \partial F(x_k + \alpha d_k) \rangle .$$

Then, for every $g(x_{k+1}) \in \partial F(x_{k+1})$, by definition of α_k^* , $\langle d_k, g(x_{k+1}) \rangle = 0$. Hence, for all $g(x_{k+1}) \in \partial F(x_{k+1})$,

$$\begin{aligned} \langle d_{k+1}, g(x_{k+1}) \rangle &= \langle s_{k+1} + \beta_k d_k, g(x_{k+1}) \rangle \\ &= \langle s_{k+1}, g(x_{k+1}) \rangle < 0 , \end{aligned} \quad (16)$$

as s_{k+1} is a descent direction.

We assume now that s_k is uniformly subgradient related. Let $\{x_k\}_{k \in K}$ a subsequence of $\{x_k\}$ such that $\lim_{k \rightarrow +\infty, k \in K} x_k = \tilde{x}$ and $0 \notin \partial F(\tilde{x})$.

As s_k is uniformly subgradient related, we directly have from Eq.(16) that $0 < \liminf_{k \rightarrow +\infty, k \in K} |F'(x_k; d_k)|$.

Moreover, as $\lim_{k \rightarrow +\infty} |\beta_k| < 1$, we have $\lim_{k \rightarrow +\infty, k \in K} \|d_k\| < +\infty$. Then d_k is uniformly subgradient related ■

However, as we do not usually have access to the optimal step, it would be interesting to know when the conjugacy parameter β_k assures to obtain an descent direction. Inspired by Al-Baali's result [1], we provide the following theorem.

Theorem 3. *Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a convex function. Let $\{x_k\}$ be a sequence generated by the conjugate descent algorithm 5, where for all k , the step size α_k was chosen under the strong Wolfe conditions (10), (12). Let $d_k = s_k + \beta_k d_{k-1}$, such that s_k is uniformly subgradient related. Let $0 < b < 1$, if $|\beta_k| < \min \left(\frac{|g_s(x_k; s_k), s_k|}{|(g_s(x_{k-1}; s_{k-1}), d_{k-1})|}, b \right)$, then d_k is a uniformly gradient related descent direction.*

Proof. We first prove by induction that d_k is a descent direction such that

$$\langle g_s(x_k, d_k), d_k \rangle \leq \langle g_s(x_k, s_k), s_k \rangle , \quad (17)$$

distinguish two cases.

1. If $\langle g_s(x_{k+1}, d_{k+1}), d_k \rangle \leq 0$, then conclusion follows immediately.
2. If $\langle g_s(x_{k+1}, d_{k+1}), d_k \rangle > 0$, then

$$|\langle g_s(x_{k+1}, d_{k+1}), d_k \rangle| \leq |\langle g_s(x_{k+1}, d_k), d_k \rangle| ,$$

and, with the strong Wolfe condition (12)

$$|\langle g_s(x_{k+1}, d_{k+1}), d_k \rangle| \leq -c_2 \langle g_s(x_k, d_k), d_k \rangle .$$

Thus

$$\frac{\langle g_s(x_{k+1}, d_{k+1}), d_{k+1} \rangle}{|\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle|} = \frac{\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle}{|\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle|} + \beta_{k+1} \frac{\langle g_s(x_{k+1}, d_{k+1}), d_k \rangle}{|\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle|} .$$

Consequently

$$\begin{aligned} \frac{\langle g_s(x_{k+1}, d_{k+1}), d_{k+1} \rangle}{|\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle|} &\leq -1 - c_2 \beta_{k+1} \frac{\langle g_s(x_k, d_k), d_k \rangle}{|\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle|} \\ &\leq -1 - c_2 \frac{\langle g_s(x_k, d_k), d_k \rangle}{|\langle g_s(x_k, s_k), d_k \rangle|} . \end{aligned}$$

By definition of $g_s(x_k, d_k)$ we have that $-1 \leq \frac{\langle g_s(x_k, d_k), d_k \rangle}{|\langle g_s(x_k, s_k), d_k \rangle|}$ and finally,

$$\frac{\langle g_s(x_{k+1}, d_{k+1}), d_{k+1} \rangle}{|\langle g_s(x_{k+1}, s_{k+1}), s_{k+1} \rangle|} \leq -1 + c_2 < 0 ,$$

which leads to $\langle g_s(x_k, d_k), d_k \rangle \leq \langle g_s(x_k, s_k), s_k \rangle$.

Let $\{x_k\}_{k \in K}$ be a subsequence of $\{x_k\}$ such that $\lim_{k \rightarrow +\infty, k \in K} x_k = \tilde{x}$ and $0 \notin \partial F(\tilde{x})$. On one hand, in a similar manner as in the proof of Lemma 1, we directly have from Eq.(17) that $0 < \liminf_{k \rightarrow +\infty, k \in K} |F'(x_k; d_k)|$.

On the other hand, as by assumption we have $\lim_{k \rightarrow +\infty} |\beta_k| < 1$ we can conclude that $\lim_{k \rightarrow +\infty, k \in K} \|d_k\| < +\infty$. Then d_k is uniformly subgradient related. ■

Note that in the smooth case, the bound on β_k reduces to the conjugate parameter proposed by Fletcher and Reeves, in which case Theorem 3 corresponds to Al-Baali's results.

5 Proximal conjugate algorithm

This section is dedicated to the proposed proximal conjugate algorithm to find a minimizer of Problem (1). We give a practical choice to choose an appropriate descent direction, thanks to the proximity operator. We begin with a study of the algorithm and show that it is an authentic conjugate gradient algorithm when f_2 is a quadratic function. We also analyze its asymptotic speed of convergence.

5.1 The algorithm

The idea is to construct a conjugate direction, based on the descent $p_k - x_k$. This gives the following algorithm:

Algorithm 7 (Proximal Conjugate Algorithm). *Initialization:* choose $x_0 \in \mathbb{R}^N$.

Repeat until convergence:

1. $p_k = \text{prox}_{f_2/L}(x_k - \frac{1}{L}\nabla f_1(x_k))$
2. $s_k = p_k - x_k$
3. Choose the conjugate parameter β_k
4. $d_k = s_k + \beta_k d_{k-1}$
5. Choose the step length α_k
6. $x_{k+1} = x_k + \alpha_k d_k$

First, we prove that the descent direction s_k provided by the proximal operator is uniformly subgradient related.

Proposition 3. *Let F be a convex function, defined as in Eq. (1) under Assumption 1, $\{x_k\}$ be a sequence generated by a descent method, $p_k = \text{prox}_{\frac{1}{L}f_2}(x_k - \frac{1}{L}\nabla f_1(x_k))$ and $s_k = p_k - x_k$. Then the sequence $\{s_k\}$ is uniformly subgradient related.*

Proof. Let \tilde{x}_k a convergent subsequence of \tilde{x} such that $\lim_{k \rightarrow \infty} \tilde{x}_k = \tilde{x}$, $\tilde{p}_k = \text{prox}_{f_2/L}(\tilde{x}_k - \frac{1}{L}\nabla f_1(\tilde{x}_k))$, and $\lim_{k \rightarrow \infty} \tilde{p}_k = \tilde{p}$. We also denote $\tilde{s}_k = \tilde{p}_k - \tilde{x}_k$ and $\lim_{k \rightarrow \infty} \tilde{s}_k = \tilde{s}$. Assume that \tilde{x} is not a critical point of F .

On one hand, we immediately have $\lim_{k \rightarrow +\infty, k \in K} \|s_k\| < +\infty$.

On the other hand, we first prove that, if x is a critical point of $F^{sur}(\cdot, x_k)$ defined in (8), then for all $h \in \mathbb{R}^N$

$$F^{sur}(x+h, x_k) - F^{sur}(x, x_k) \geq \frac{L}{2} \|h\|_2^2.$$

For that, we compute $\partial_x F^{sur}(x, a)$:

$$\partial_x F^{sur}(x, a) = \nabla f_1(x) + L(x - a) + \partial f_2(x),$$

and define:

$$g_s^{sur}(x, a; d) = \arg \sup_{g \in \partial_x F^{sur}(x, a)} \langle g, d \rangle.$$

As a consequence $g_s^{sur}(x, x; d) = g_s(x; d)$. One can check that

$$\begin{aligned} F^{sur}(x+h, x_k) - F^{sur}(x, x_k) &= \langle \partial F^{sur}(x, x_k), h \rangle + L/2 \|h\|_2^2 \\ &\quad + \{f_2(x+h) - f_2(x) - \langle \partial f_2(x), h \rangle\}. \end{aligned}$$

Since x is a critical point of $F^{sur}(\cdot, x_k)$, for all h , we have $\langle \partial F^{sur}(x, x_k), h \rangle = 0$, then

$$F^{sur}(x+h, x_k) - F^{sur}(x, x_k) = L/2\|h\|_2^2 + \{f_2(x+h) - f_2(x) - \langle \partial f_2(x), h \rangle\} .$$

By definition of the subgradient, an element v belongs to $\partial f_2(x)$ if and only if for all y , $f_2(x) + \langle v, y-x \rangle \leq f_2(y)$. In particular, when $y = x+h$, for all h and for all $v \in \partial f_2(x)$, we have that

$$f_2(x) + \langle v, h \rangle \leq f_2(y) \text{ i.e. } 0 \leq f_2(x+h) - f_2(x) - \langle \partial f_2(x), h \rangle ,$$

and

$$F^{sur}(x+h, x_k) - F^{sur}(x, x_k) \geq L/2\|h\|_2^2 .$$

Now, we apply the previous inequality to $x = p_k$, which is a critical point of $F^{sur}(\cdot, x_k)$ as seen in Section 3.2, and to $h = -s_k$. This gives

$$\begin{aligned} -L/2\|s_k\|^2 &\geq F^{sur}(p_k, x_k) - F^{sur}(p_k - s_k, x_k) \\ &\geq F^{sur}(p_k, x_k) - F^{sur}(x_k, x_k) \\ &\geq \langle g_s^{sur}(x_k, x_k; s_k), s_k \rangle \\ &\geq \langle g_s(x_k; s_k), s_k \rangle , \end{aligned}$$

where the third inequality comes from the definition of the subgradient $g_s^{sur}(x_k, x_k; s_k)$, for the descent direction $s_k = p_k - x_k$. Taking the limit, we have then

$$L/2\|\tilde{s}\|^2 \leq \liminf |\langle g_s(\tilde{x}, \tilde{s}), \tilde{s} \rangle| .$$

As $\tilde{s} \neq 0$ (otherwise, \tilde{x} is a critical point), the proposition follows . \blacksquare

Then, if α_k is chosen with the Wolfe conditions, the proximal conjugate algorithm converges (assuming that β_k is chosen so that d_k is still a descent direction for all k). Furthermore, if α_k is chosen with the Mifflin-Wolfe conditions, we also have the convergence of the algorithm thanks to Theorem 2.

5.2 Remarks on the step size

Variants of ISTA estimate at each iteration the Lipschitz-parameter L in order to ensure convergence of the Algorithm. Such a variant is restated in Algorithm 8. One can refer for example to [3] for more details.

Algorithm 8 (ISTA with Line search). *Initialization:* choose $x_0 \in \mathbb{R}^N$ and $\eta > 1$.

Repeat until convergence:

1. Find the smallest integer i_k such that with $\mu_k = \frac{1}{\eta^{i_k} L_{k-1}}$ and with

$$x_{k+1} = \text{prox}_{\mu_k f_2}(x_k - \mu_k \nabla f_1(x)) ,$$

we have $F(x_{k+1}) \leq \bar{F}^{sur}(x_{k+1}, x_k)$, where \bar{F}^{sur} is defined as in Eq. (8) replacing L by $\eta^{i_k} L_{k-1}$.

Then, in frameworks like SPARSA [33], the authors propose to use μ_k as a step parameter, and propose strategies as the Bazilei-Borwein choice to set it up. The following lemma establishes a necessary and sufficient condition which states that μ_k is equivalent to the step-size parameter α_k in Algorithm 7 (when the conjugate parameter β_k is set to zero).

Lemma 2. *Let F be a convex function defined as in Eq. (1) under Assumption 1, $p_k = \text{prox}_{\frac{1}{L}f_2}(x_k - \frac{1}{L}\nabla f_1(x_k))$, $x_{k+1} = x_k + \alpha_k(p_k - x_k)$. We also have $x_{k+1} = \text{prox}_{\frac{\alpha_k}{L}f_2}(x_k - \frac{\alpha_k}{L}\nabla f_1(x_k))$ if and only if $\partial f_2(p_k) \cap \partial f_2(x_{k+1}) \neq \emptyset$.*

Proof. By definition of the proximity operator, $x_k - \frac{1}{L}\nabla f_1(x_k) - p_k \in \frac{1}{L}\partial f_2(p_k)$.

Let us denote by $p_k^\alpha = \text{prox}_{\frac{\alpha_k}{L}f_2}(x_k - \frac{\alpha_k}{L}\nabla f_1(x_k))$. Then

$$\begin{aligned} p_k^\alpha = x_k + \alpha_k(p_k - x_k) &\Leftrightarrow x_k - \frac{\alpha_k}{L}\nabla f_1(x_k) - x_k - \alpha_k(p_k - x_k) \in \frac{\alpha_k}{L}\partial f_2(p_k^\alpha) \\ &\Leftrightarrow 0 \in -\frac{\alpha_k}{L}\nabla f_1(x_k) + \frac{\alpha_k}{L}(\nabla f_1(x_k) + \partial f_2(p_k)) - \frac{\alpha_k}{L}\partial f_2(p_k^\alpha) \\ &\Leftrightarrow 0 \in \partial f_2(p_k) - \partial f_2(p_k^\alpha) \\ &\Leftrightarrow \partial f_2(p_k) \cap \partial f_2(x_{k+1}) \neq \emptyset \end{aligned}$$

■

However, the necessary and sufficient condition given in the previous Lemma is hard to check, and can never occur for certain choices of function f_2 (for example, if f_2 is differentiable).

5.3 The quadratic case

A natural question concerns the behavior of this proximal-conjugate descent algorithm when f_2 is quadratic, i.e.

$$f_2(x) = \frac{1}{2}\langle x, Qx \rangle + \langle c, x \rangle,$$

with Q a symmetric definite positive linear application, and $c \in \mathbb{R}^N$. We have then

$$\begin{aligned} \hat{x} = \text{prox}_{\mu f_2}(y) &= \arg \min_x \frac{1}{2}\|y - x\|^2 + \mu f_2(x) \\ \Leftrightarrow 0 &= \hat{x} - y + \mu Qx + \mu c \\ \Leftrightarrow \hat{x} &= (I + Q\mu)^{-1}(y - \mu c) \end{aligned}$$

Hence, the descent direction s_k given in the proximal conjugate algorithm is

$$\begin{aligned} s_k &= \text{prox}_{\mu f_2}(x_k - \mu\nabla f_1(x_k)) - x_k \\ &= (I + \mu Q)^{-1}(x_k - \mu\nabla f_1(x_k) - \mu c) - x_k \\ &= (I + \mu Q)^{-1}(-\mu\nabla f_1(x_k) - \mu c - \mu Qx_k) \\ &= -\left(\frac{1}{\mu}I + Q\right)^{-1}(\nabla f_1(x_k) + \nabla f_2(x_k)) \end{aligned}$$

The proximal conjugate descent is then the classical conjugate gradient algorithm preconditioned by $\frac{1}{\mu}I + Q$.

5.4 Speed of convergence

Intuitively, the conjugate algorithm has asymptotically the same behavior as ISTA. Then, one can expect that the speed of convergence will be $O(1/k)$, for k large enough. This is stated with the following theorem.

Theorem 4. *Let F be a convex function satisfying Assumption 1 and x^* a minimizer of F . Let $\{x_k\}$ be the sequence generated by the proximal conjugate Algorithm 7. Then, there exist $K > 0$ such that for all $k > K$, $F(x_k) - F(x^*) \leq \frac{L\|x^* - x_k\|^2}{2(k-K+1)}$.*

Proof. The proof is based on the one given by Tseng in [30] for the speed of convergence of ISTA.

Let

$$\ell_F(x; y) = f_1(y) + \langle \nabla f_1(y), x - y \rangle + \lambda f_2(x) .$$

We can recall the “three points property”: if $z_+ = \arg \min_x \psi(x) + \frac{1}{2}\|x - z\|^2$, then

$$\psi(x) + \frac{1}{2}\|x - z\|^2 \geq \psi(z_+) + \frac{1}{2}\|z_+ - z\|^2 + \frac{1}{2}\|x - z_+\|^2$$

Moreover, with the following inequality

$$F(x) \geq \ell_F(x; y) \geq F(x) - \frac{L}{2}\|x - y\|^2 ,$$

$$\begin{aligned} F(p_k) &\leq F(x) + \frac{L}{2}\|x - x_k\|^2 - \frac{L}{2}\|x - p_k\|^2 \\ \sum_{n=K}^k F(p_n) - F(x) &\leq \frac{L}{2} \sum_{n=K}^k k(\|x - x_n\|^2 - \|x - p_n\|^2) \end{aligned}$$

Since the sequence of $F(p_k)$ is decreasing, we have

$$\begin{aligned} (k - K + 1)(F(p_k) - F(x)) &\leq \frac{L}{2} \sum_{n=K}^k (\|x - x_n\|^2 - \|x - p_n\|^2) \\ &\leq \frac{L}{2} \sum_{n=K}^k (\|x - x_n\|^2 - \|x - x_{n+1}\|^2 - \|x_{n+1} - p_n\|^2) \\ &\leq \frac{L}{2}\|x - x_k\|^2 - \frac{L}{2}\|x - x_{k+1}\|^2 - \frac{L}{2} \sum_{n=K}^k \|x_{n+1} - p_n\|^2 \\ &\leq \frac{L}{2}\|x - x_k\|^2 - \frac{L}{2} \sum_{n=K}^k \|x_{n+1} - p_n\|^2 \end{aligned}$$

For all ε_1 , there exists a number K_1 for which all $k \geq K_1$ $|F(x_k) - F(p_k)| < \varepsilon_1$. Moreover, for all ε_2 , there exists a number K_2 such that for all $k \geq K_2$ $\|x_{k+1} - p_k\| < \varepsilon_2$. The choices $\varepsilon_1 = \frac{L}{2}\varepsilon_2$ and $K = \max(K_1, K_2)$, ensure that for all $k > K$

$$F(x_k) - F(x^*) \leq \frac{L\|x^* - x_k\|^2}{2(k - K + 1)} - \frac{L}{2}\varepsilon_2 + \varepsilon_1$$

$$F(x_k) - F(x^*) \leq \frac{L\|x^* - x_k\|^2}{2(k - K + 1)} .$$

■

5.5 An approximate proximal conjugate descent algorithm

In Algorithm 7, one must be able to compute exactly the proximity operator of function f_2 . However, in many cases, one does not have access to a close form solution, but can only approximate it thanks to iterative algorithms. In that case, a natural question arises: how does behave the proposed algorithm when we cannot have a close form formula for the proximity operator?

The study made in Section 4 shows that one needs to obtain a descent direction s_k to construct the conjugate direction d_k . Remember that the proximity operator has exactly the form of the general optimization problem given by Eq. (1). Then, any iterative algorithm able to deal with this kind of problem can estimate the solution of the proximity operator, within an inner loop of the main proximal conjugate algorithm.

Using such a procedure may be computationnaly costly. Nevertheless, with a few iterations of the inner loop, the functional decreases. Since we only need a descent direction, as defined in Definition 3, we are looking for an algorithm where step 1. in Algorithm 7 is replaced by:

1. Find \check{p}_k such that $F^{sur}(\check{p}_k, x_k) < F^{sur}(x_k, x_k)$

Indeed, in that case we have

$$F(\check{p}_k) = F^{sur}(\check{p}_k, \check{p}_k) \leq F^{sur}(\check{p}_k, x_k) \leq F^{sur}(x_k, x_k) = F(x_k) ,$$

regarding the definition of the surrogate F^{sur} given by Eq. (8) and the inequality (9). Then at Step 2 of the proximal conjugate algorithm, $s_k = \check{p}_k - x_k$ is guaranteed to be a descent direction. But, this descent direction may not be uniformly subgradient related anymore and there is no more guarantee to converge to a minimizer of the functional. Nevertheless for a certain class of function f_2 , we can establish a strategie which ensure the convergence. From now, we assume the following.

Assumption 2. *There exists a linear operator $\Phi : \mathbb{R}^N \rightarrow \mathbb{R}^M$ and a function $\tilde{f} : \mathbb{R}^M \rightarrow \mathbb{R}^M$ such that $f_2 : \mathbb{R}^N \rightarrow \mathbb{R}$ can be written as*

$$\mu f_2(x) = \tilde{f}(\Phi x) .$$

Denoting by \tilde{f}^* , the Fenchel conjugate of \tilde{f} , we suppose that the proximity operator of \tilde{f}^* is given by a closed form.

Again, we do not have access to a closed form for $\text{prox}_{\mu f_2}$. However, using the Fenchel dual formulation we can rewrite this minimization problem such that

$$\min_u \frac{1}{2} \|y - u\|_2^2 + \tilde{f}(\Phi u) = \max_v -\|\Phi^* v\| - \langle \phi^* v, y \rangle + \tilde{f}^*(v) .$$

Moreover, thanks to the KKT conditions, the following relationship between the primal variable u and the dual variable v holds:

$$u^* = y + \Phi^* v^* .$$

Hence, one can use any known algorithm to obtain an approximation of the proximal solution at step 1 of Algorithm 7. Such a strategy is already used in practice (see for example [12, 2]). However, this inner loop is usually run in order to obtain an estimate close to the true minimizer, and may be a computational burden. In the light of the remark above, we propose to stop the inner loop as soon as a point allowing to decrease the original functional is obtained. This strategy is summarized in the following algorithm, where one can use any first order algorithm in the inner loop.

Algorithm 9 (Approximate Proximal Conjugate Algorithm). *Initialization:* choose $x_0 \in \mathbb{R}^N$

Repeat until convergence:

1. $y_k = x_k - \frac{1}{L} \nabla f_1(x_k)$
2. Computation of p_k such that $F^{sur}(p_k, x_k) \leq F^{sur}(x_k, x_k)$, by solving the dual problem of $\min_p \frac{1}{2} \|y_k - p\|_2^2 + \frac{\lambda}{L} f_2(p)$
3. $s_k = x_k - p_k$
4. Choose the conjugate parameter β_k
5. $d_k = -s_k + \beta_k d_{k-1}$
6. Choose the step length α_k
7. $x_{k+1} = x_k + \alpha_k d_k$

When β_k is set to zero at each iteration, the step size α_k is set to one and the inner loop is run until “convergence”. In the latter case the algorithm reduced to the one proposed for the Total Variation regularized inverse problems in [12]. Here, we propose a simple criterion to stop the inner loop, and the convergence is given by the following theorem.

Theorem 5. *Let $\{x_k\}$ be a sequence generated by Algorithm 9. Assume that for all k , d_k is a descent direction and β_k is bounded. Then, if α_k is chosen thanks to the Mifflin-Wolfe conditions, or is a constant step size, $\{x_k\}$ converges to a minimizer of F .*

Proof. We first show that, in a finite number of iterations, we can find $p_k = y + \Phi^* v_k$, such that $F^{sur}(p_k, x_k) < F^{sur}(x_k, x_k)$, if x_k is not a minimizer of $F^{sur}(\cdot, x_k)$. Assume the opposite: $\forall \ell F^{sur}(p_\ell, x_k) \geq F^{sur}(x_k, x_k)$. Then the sequence of dual variable v_ℓ generated by the inner loop converges to a fixed point of $\text{prox}_{\tilde{f}}(\frac{1}{2}\|\Phi^* \cdot\|_2^2 + \langle y_k, \Phi^* \cdot \rangle)$, and by definition of the Fenchel duality, p_ℓ converges to $\arg \min_p \frac{1}{2}\|y_k - p\|^2 + \lambda f_2(p)$. Hence $\lim_{\ell \rightarrow \infty} F^{sur}(p_\ell, x_k) = F^{sur}(x_k, x_k)$, contradicting that x_k is not a minimizer of $F^{sur}(\cdot, x_k)$.

Secondly, using the same arguments as in Theorem 2, we have $\lim_{k \rightarrow 0} \|d_k\| = 0$, and then $\lim_{k \rightarrow 0} \|s_k\| = 0$. Let \tilde{x} be an accumulation point of $\{x_k\}$, which is also an accumulation point of $\{p_k\}$. We have

$$\begin{aligned} \lim_{k \rightarrow \infty} F^{sur}(p_k, x_k) &= F^{sur}(\tilde{x}, \tilde{x}) \\ &= \min_p F^{sur}(p, \tilde{x}) \\ &= \min_x F(x) \quad \text{by definition of } F^{sur}. \end{aligned}$$

Then, applying Theorem 1, Algorithm 9 converges. ■

5.6 Stopping criterion

We discuss here a strategy based on the computation of *duality gaps* to derive principled stopping criterion for the previous algorithms. When the cost function \mathcal{F} is smooth, a natural optimality criterion is obtained by checking that the gradient is 0. The condition reads $\|\nabla \mathcal{F}(X^{(k)})\| < \varepsilon$. Unfortunately, cost-functions involving ℓ_1 norms are non-differentiable, and looking at the norm of the sub-gradients does not help.

When considering convex problem, a solution is to compute the “duality-gap”, if possible. Based on the Fenchel-Rockafellar [27] duality theorem, it is known that for the problems we consider the gap at the optimum is 0.

Theorem 6 (Fenchel-Rockafellar duality [27]). *Let $f : \mathbb{R}^M \cup \{+\infty\} \rightarrow \mathbb{R}$ be a convex function and $g : \mathbb{R}^N \cup \{+\infty\} \rightarrow \mathbb{R}$ a concave function. Let G be a linear operator mapping vectors of \mathbb{R}^M to \mathbb{R}^N . Then*

$$\inf_{x \in \mathbb{R}^M} \{f(x) - g(Gx)\} = \sup_{y \in \mathbb{R}^N} \{g^*(y) - f^*(G^* y)\}$$

where f^* (resp. g^*) is the Fenchel conjugate associated to f (resp. g), and G^* the adjoint operator of G .

Moreover, the Karush-Kuhn-Tucker (KKT) conditions read:

$$f(X) + f^*(G^* u) = \langle x, G^* y \rangle, \quad g(Gx) + g^*(y) = \langle Gx, y \rangle.$$

The duality gap is then define as $\eta_k = |f(x_k) - g(Gx_k) - g^*(y_k) - f^*(G^* y_k)|$, where the mapping between x_k and y_k is given by the KKT conditions.

Such a criterion is discussed for example in [17].

6 Numerical illustrations

We provide in this section two experiences to show the behavior of the presented algorithms 7 and 9, denoted by ProxConj in the following. These experiments are made on the block signal, displayed on Figure 6, used in several papers of Donoho (see for example [6]), which has a length of 1024 samples.

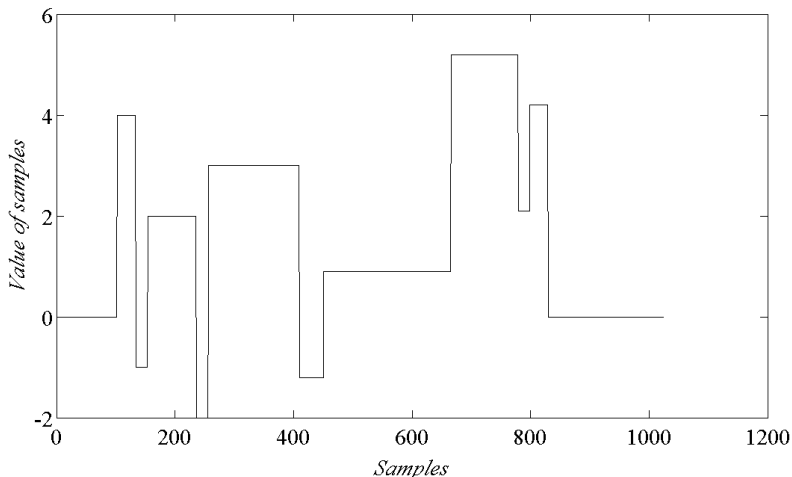


Figure 1: The block signal

The functionals we minimize are constructed using a “compressed sensing” framework [11],[7]. Denoted by s the original block signal, we apply a random sensing matrix A , and then add a white Gaussian noise b to obtain the observed signal $y = As + b$. The random matrix A is generated using normalized centered Gaussian random vectors, and the white Gaussian noise has a standard deviation $\sigma_0 = 15$.

Let us stress that this Section is provided to support the discussion made in the next section instead of discuss the performance of the algorithms on a particular application.

6.1 Experiment on a synthesis problem

The first experiment use the fact that the signal s is sparse in a wavelet dictionary. We then choose a Haar wavelet basis, and we seek to minimize

$$\frac{1}{2} \|y - A\Phi x\|_2^2 + \lambda \|x\|_1$$

where Φ is the matrix associated with the Haar wavelet basis. λ is chosen in order to reach approximately the best Signal to Noise Ratio between the original signal and the estimated one $\hat{s} = \Phi\hat{x}$, with \hat{x} the computed minimizer ($\lambda = 500$).

We compare the performance of the following algorithms:

- FISTA;
- ISTA;
- ISTA with an optimal step length;
- ProxConj with an optimal step length, where the optimal step is computed thanks to (expensive) numerical optimization. The conjugate parameter is chosen as $\beta_k = \max(0, \frac{\langle s_k - s_{k-1}, s_k \rangle}{\|s_{k-1}\|_2^2})$.
- ProxConj with the Wolfe-Mifflin line search of the step length. The conjugate parameter is chosen as above, but we check at each iteration if the functional value decrease.

We display on Figure 6.1 the evolution of the functionals values during the iterations.

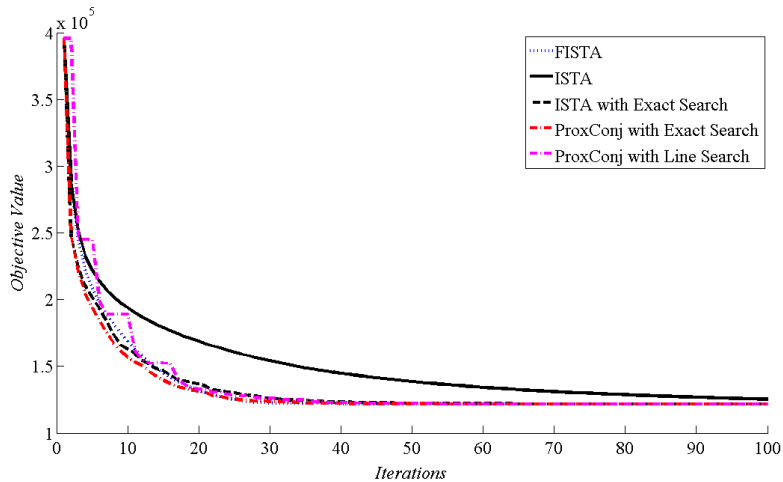


Figure 2: Comparison of different algorithms on a synthesis problem.

6.2 Experiment on a analysis problem

The second experiment use the fact that the block signal must have a small ℓ_1 norm of its total variation. We then minimize the following functional:

$$\frac{1}{2} \|y - Ax\|_2^2 + \lambda \|D^T x\|_1 ,$$

where D is a finite difference operator (hence $\|D^T \bullet\|_1$ correspond to a discrete Total Variation penalization). As in the previous experiment, λ is chosen to

maximize the SNR of the estimated signal ($\lambda = 1000$). We compare the same algorithms, which share the same strategie to stop the inner loop. Figure 6.2 shows the evolution of the functional values during the iterations.

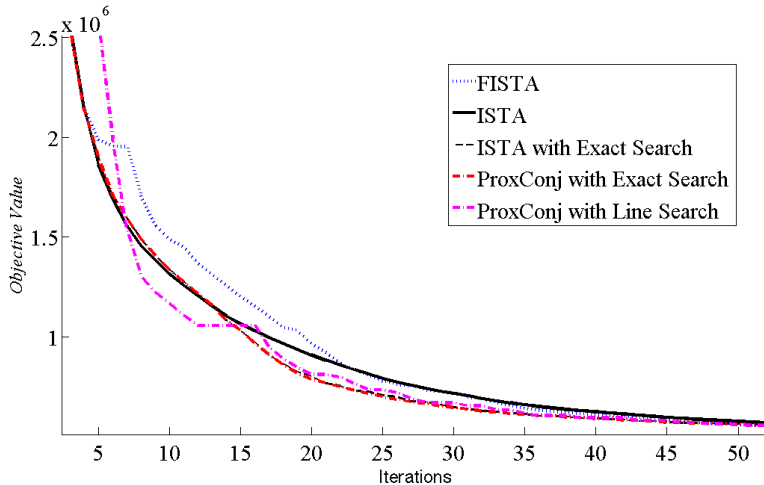


Figure 3: Comparison of different algorithms on an analysis problem.

Let us stress that the curves show the functionals values with respect to the number of iterations, not with respect to the CPU time. In terms of computational time, FISTA remains the faster algorithm (thanks to the simplicity of each iteration). Next section provide a more detailed discussion about the shortcomings, but also some hopes, of the proposed algorithms.

7 Discussion

The main goal of this contribution was to answer the following question: *as the conjugate gradient algorithm is popular for differentiable functions, is it possible to adapt it to non-differentiable ones ?* As the proximal algorithm is able to find a descent direction, it seems natural to try to “conjugate” them during the iteration. The study made in this contribution is mainly theoretical, and there is still some issues in order to use the proximal-conjugate algorithms in practice.

In particular, the choice of the step length is certainly the most difficult, and one can spent a lot of time in order to choose an adequate step length. A good choice of this step can greatly increase the speed of convergence of an algorithm: the proximal conjugate algorithm, and also ISTA, with an optimal step length give particularly good results. However, computation of an optimal step length is usually avoided in practice if no closed form is provided. The Mifflin-Wolfe conditions give a practical way to obtain a step length. However, the optimal step length does not necessarily satisfy these conditions. In the

previous experiments, the step was sometimes very small and did not decrease the functional value significantly.

Another shortcoming of the proposed conjugate algorithm, is the choice of the conjugate parameter β_k during the iterations. The choice made in the experiments does not guarantee to obtain a descent direction at each iteration. Moreover, the sufficient condition given by theorem 3 is actually difficult to check in practice.

Finally, the asymptotical speed of convergence of the algorithm is slower than the one of FISTA. However, the experiments show that during the first iterations, the functional decrease very quickly compared to FISTA.

In the future, it would be interesting to find a efficient strategy in order to choose a “good” step length. Moreover, one should investigate the possible and efficient choices of the conjugate parameter β_k , as it was done for the conjugate gradient decent, in particular to be sure that the resulting direction is a descent direction. Last but not least, the question of the generalization in the case of non-convex functional remains open.

Acknowledgement

The author warmly thanks Aurélia Fraysse and Pierre Weiss for fruitful discussion.

References

- [1] M. Al-Baali. Descent property and global convergence of the fletcher-reeves method with inexact line search. *IMA Journal of Numerical Analysis*, 5:121–124, 1985.
- [2] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.
- [3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- [4] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.
- [5] Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal, and Claudia A. Sagastiábal. *Numerical Optimization*. Springer, 2003.
- [6] J. Buckheit and D. L. Donoho. *Wavelets and Statistics*, chapter Wavelet and reproducible research. Springer-Verlag, Berlin, New York, 1995.

- [7] E. J. Candès and T. Tao. Near optimal signal recovery from random projections : universal encoding strategies ? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.
- [8] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.
- [9] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, November 2005.
- [10] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math.*, 57(11):1413 – 1457, Aug 2004.
- [11] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [12] Jalal Fadili and Gabriel Peyré. Total variation projection with first order schemes. Technical report, 2009.
- [13] R. Fletcher and C. Reeves. Function minimization by conjugate gradients. *Comput. Journal*, 7:149–154, 1964.
- [14] William G. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization - Special Issue on Conjugate Gradient and Quasi-Newton Methods for Nonlinear Optimization*, 2(1):35 – 58, Jan 2006.
- [15] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
- [16] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer-Verlag, 1993.
- [17] S.-J Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.
- [18] R. Mifflin. An algorithm for constrained optimization with semismooth functions. *Math. Oper. Res.*, 2:191 – 207, 1977.
- [19] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.
- [20] Y.E. Nesterov. Gradient methods for minimizing composite objective function. Technical report, 2007. CORE discussion paper – Université Catholique de Louvain.

- [21] Yurii E. Nesterov. method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.
- [22] M. Ng, P. Weiss, and X.-M. Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. Technical report, 2009.
- [23] E. Polak and Ribière. Note sur la convergence de directions conjuguées. *Revue Française d’Informatique et de Recherche Opérationnelle*, 3(16):35–43, 1969.
- [24] B.T. Polyak. The conjugate gradient method in extreme problems. *USSR Comp. Math. Math. Phys.*, 9:94–112, 1969.
- [25] B.T. Polyak. *Introduction to Optimization*. Translation Series in Mathematics and Engineering, Optimization Software, 1987.
- [26] Radoslaw Pytlak. *Conjugate Gradient Algorithms in Nonconvex Optimization*. Springer, 2009.
- [27] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1972.
- [28] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. 1994.
- [29] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Serie B*, 58(1):267–288, 1996.
- [30] Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. Technical report, 2009.
- [31] Pierre Weiss. *Algorithmes rapides d’optimisation convexe. Applications à la reconstruction d’images et à la détection de changements*. PhD thesis, Université de Nice Sophia-Antipolis, Novembre 2008.
- [32] Philip Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Studies*, 3:145–173, 1975.
- [33] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.
- [34] Jin Yu, S.V.N. Vishwanathan, Simon Gunter, and Schraudolph Nicol N. A quasi-newton approach to nonsmooth convex optimization problems in machine learning. *Journal of Machine Learning Research*, 11:1 – 57, 2010.