Variable Metric Forward-Backward Algorithm for Composite Minimization Problems

Audrey Repetti† and Yves Wiaux⋆

† School of Mathematics and Computer Sciences, Heriot-Watt University, Edinburgh, UK
⋆ School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, UK
{a.repetti, y.wiaux}@hw.ac.uk

Abstract

We present a forward-backward-based algorithm to minimize a sum of a differentiable function and a nonsmooth function, both being possibly nonconvex. The main contribution of this work is to consider the challenging case where the nonsmooth function corresponds to a sum of composite functions whose proximity operators cannot be computed efficiently. We assume that each of these functions is the composition between a strictly increasing, concave, differentiable function and a nonsmooth function whose proximity operator can be computed efficiently. The proposed variable metric Composite Function Forward-Backward algorithm (C2FB) circumvents the explicit computation of the proximity operator of the composite functions through a majorize-minimize approach. Precisely, each composite function is majorized using a linear approximation of the differentiable function, which allows one to apply the proximity step only to the sum of the nonsmooth functions. We prove the convergence of the algorithm iterates to a critical point of the objective function leveraging the Kurdyka-Łojasiewicz inequality. The convergence is guaranteed even if the proximity operators are computed inexactly, considering relative errors. We show that the proposed approach is a generalization of reweighting methods, with convergence guarantees. In particular, applied to the log-sum function, our algorithm reduces to a generalized version of the celebrated reweighted ℓ₁ method. Finally, we show through simulations on an image processing problem that the proposed approach accelerates traditional reweighting methods.

Keywords. Nonconvex optimization, nonsmooth optimization, proximity operator, composite minimization problem, majorize-minimize method, forward-backward algorithm, reweighting algorithm, inverse problems

MSC. 90C26, 90C59, 65K10, 49M27, 68W25, 68U10, 94A08
1 Introduction

In this work we consider optimization problems of the form

\[
\minimize_{x \in \mathbb{R}^N} \left\{ f(x) = h(x) + g(x) \right\},
\]

(1.1)

where \( h : \mathbb{R}^N \to \mathbb{R} \) is a Lipschitz differentiable function with constant \( \mu > 0 \), and \( g : \mathbb{R}^N \to [-\infty, +\infty] \) is a composite function as follows

\[
(\forall x \in \mathbb{R}^N) \quad g(x) = \phi \circ \psi(x).
\]

(1.2)

Composite optimization problems of this form have been studied extensively during the last decades, notably in, e.g., [10, 12, 22, 24, 30, 39, 40, 47, 49]. In general, all these works rely on the same strategy, consisting in iteratively minimizing approximations to the objective function \( f \).

1.1 Related work

A first notable example is when \( \phi \) is the identity function and \( g \equiv \psi \) is a proper, lower semi-continuous function whose proximity operator\(^1\) can be computed. In this context, a common approach to solve (1.1) is the forward-backward (FB) algorithm [13, 20, 32, 46], which alternates between a gradient step on \( h \) and a proximity step on \( g \). Precisely, at each iteration \( k \in \mathbb{N} \), given the current iterate \( x_k \in \mathbb{R}^N \), the next iterate is defined as

\[
x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla h(x_k)),
\]

(1.3)

where \( \text{prox}_{\gamma_k g} \) denotes the proximity operator of \( \gamma_k g \), and \( \gamma_k > 0 \). The convergence of the iterates \( (x_k)_{k \in \mathbb{N}} \) of the FB algorithm to a minimizer of \( f \) has been established when both \( h \) and \( g \) are convex, and choosing \( \gamma_k \in (0, 2/\mu] \) (see e.g. [9, 20]). This result has been extended in [2] to the case when both \( h \) and \( g \) are nonconvex. Assuming that \( \gamma_k \in (0, 1/\mu] \), the authors have proved the convergence of \( (x_k)_{k \in \mathbb{N}} \) to a critical point of \( f \), using the Kurdyka-Łojasiewicz (KL) inequality [6, 29, 33]. However, as many first-order minimization methods, it may suffer from slow convergence [13].

Recently, accelerated versions of the FB algorithm have been proposed in the literature, mainly based either on Nesterov’s accelerations [3, 31] (also called subspace accelerations, or FISTA) or on preconditioning strategies [15, 16, 25]. While the former uses information from the previous iterates, the second one aims to improve the step-size \( \gamma_k \) at each iteration by introducing a symmetric positive definite (SPD) matrix \( A_k \in \mathbb{R}^{N \times N} \), leading to the following iteration update (known as variable metric forward-backward algorithm (VMFB)):

\[
x_{k+1} = \text{prox}_{\gamma_k^{-1} A_k} (x_k - \gamma_k A_k^{-1} \nabla h(x_k)).
\]

(1.4)

\(^1\)The definition of the proximity operator and all the other mathematical definitions and notation used throughout this paper will be given in Section 2.
When $A_k$ is chosen to be equal to the identity matrix $I_N$, then the basic FB algorithm is recovered. However, as shown in [15, 41], wiser choices of $A_k$ can drastically accelerate the convergence of the iterates, and even outperform FISTA. In particular, in [15], the authors proposed to choose the preconditioning matrices using a majorize-minimize (MM) approach [28, 34, 45, 48]. Indeed, one can notice that (1.4) can be equivalently rewritten as an MM algorithm:

$$x_{k+1} = \arg\min_{x \in \mathbb{R}^N} \left\{ f_k(x, x_k) = g(x) + h(x_k) + \langle \nabla h(x_k) \mid x - x_k \rangle + \frac{1}{2\gamma_k} \langle x - x_k \mid A_k(x - x_k) \rangle \right\}. \quad (1.5)$$

When $A_k = I_N$, using the descent lemma, it is straightforward to notice that $f_k(\cdot, x_k) : \mathbb{R}^N \to ]-\infty, +\infty]$ is a majorant function of $f$ at $x_k$ in the sense that, for every $x \in \mathbb{R}^N$, $f(x) \leq f_k(x, x_k)$ and $f(x_k) = f_k(x_k, x_k)$. In [15], the authors proposed to choose $A_k$ to define a more accurate majorant function at each iteration, and proved the convergence of sequences $(x_k)_{k \in \mathbb{N}}$ generated by (1.4) to a critical point of $f$ using the KL inequality (when $g$ is assumed to be convex). Unfortunately, to the best of our knowledge, there exists no version of the FB algorithm with convergence guarantees able to solve the general composite problem (1.1)-(1.2) for more general choices of $\phi$, in particular when the proximity operator of $g$ cannot be computed.

During the last years, many optimization methods arose to solve the general composite problem (1.1)-(1.2), mainly based on MM strategies. In particular, we can distinguish two main approaches. The first one consists in majorizing the outer function $\phi$ in (1.2). This approach has been investigated in [26, 37]. Precisely, in [37], the authors assume that $\psi = (\psi_1, \ldots, \psi_P)$ where $P \leq N$ and, for every $p \in \{1, \ldots, P\}$, $\psi_p : \mathbb{R}^N \to \mathbb{R}$ is convex, and $\phi : \mathbb{R}^P \to \mathbb{R}$ is coordinate-wise non-decreasing. They also do not require $h$ to be differentiable, but to be proper, lower semi-continuous and convex. In this context, the authors propose to solve (1.1)-(1.2) by defining

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = \arg\min_{x \in \mathbb{R}^N} h(x) + q(\psi(x), x_k), \quad (1.6)$$

where $q : \mathbb{R}^P \to \mathbb{R}$ is a convex, proper, non-decreasing majorant function of $\phi$ at $\psi(x_k)$. Using the KL inequality, the authors show that $(x_k)_{k \in \mathbb{N}}$ converges to a critical point of $f$. However, problem (1.6) needs to be solved exactly at each iteration, which may not be possible in practice. Note that convergence of general MM algorithms in a non-convex setting (not necessarily for composite functions) has also been investigated in [7], but it necessitates as well problem (1.6) to be solved exactly at each iteration to ensure the convergence of $(x_k)_{k \in \mathbb{N}}$. In [26], the authors propose a similar approach, using the same update as in (1.6), but under different assumptions, and with a particular form for the majorant function $q(\cdot, x_k)$. Precisely, for every $k \in \mathbb{N}$, the authors propose to choose the majorant function as follows

$$(\forall u \in \mathbb{R}^P) \quad q(u, x_k) = \phi(\psi(x_k)) + \langle \nabla \phi(\psi(x_k)) \mid u - \psi(x_k) \rangle + \frac{1}{\gamma} D_d(u, \psi(x_k)), \quad (1.7)$$

where $\gamma > 0$ and $D_d$ is the Bregman distance relative to a Legendre function $d$ [43]. In particular, when $d$ is the usual Euclidean norm squared, then (1.7) corresponds to a quadratic majorant of $\phi$.
at $\psi(x_k)$. In this work, the authors do not assume $h$ to be differentiable, but only proper and lower semi-continuous. The function $\psi$ is assumed to be continuously differentiable, and $\phi$ is assumed to be differentiable such that $ad - \phi$ is convex for some constant $a > 0$, and $\gamma^{-1}\nabla d - \nabla \phi$ is locally Lipschitz continuous on int dom $d$ (for $d$ strongly convex given by the Bregman distance in (1.7)). Under these technical assumptions and using the KL inequality, the convergence of $(x_k)_{k \in \mathbb{N}}$ is then guaranteed. However, both functions $\phi$ and $\psi$ necessitate to be differentiable, and problem (1.6) must be solved accurately at each iteration (most of the time using sub-iterations). The second approach to solve the full composite problem (1.1)-(1.2) consists in using Taylor-like models, and has been investigated in [23, 38]. In both the works, the authors propose to investigate a general problem of the form

$$\min_{x \in \mathbb{R}^N} f(x).$$

(1.8)

In particular, in [23], the authors propose to define the next iterate as

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = \arg\min_{x \in \mathbb{R}^N} f(x_k)(x),$$

(1.9)

where $f(x_k) : \mathbb{R}^N \to ]-\infty, +\infty[$ is a model function for $f$ at $x_k$, in the sense that there exists a growth function $w : [0, +\infty[ \to [0, +\infty]$ such that, for every $x \in \mathbb{R}^N$, $|f(x) - f(x_k)(x)| \leq w(|x - x_k|)$. In [38], the authors propose a modified version of (1.9):

$$(\forall k \in \mathbb{N}) \quad x_{k+1} = \arg\min_{x \in \mathbb{R}^N} f(x_k)(x) + D_{d_k}(x, x_k),$$

(1.10)

where $d_k$ is a Legendre function. In both the cases, sub-iterations are needed at each iteration to solve either (1.9) or (1.10). In addition, both the two works have similar convergence guarantees, obtained without the need of the KL inequality. It is interesting to note that, in [38], the authors pointed out that (1.10) must be solved accurately to reach asymptotic convergence.

In the current work, we propose a novel algorithm merging the structure of the VMFB algorithm with the MM strategy consisting in approximating the composite function of interest.

1.2 Proposed approach

In this work, we will be particularly interested in solving problem (1.1) where $g$ is of the form

$$(\forall x \in \mathbb{R}^N) \quad g(x) = \sum_{p=1}^{P} \lambda_p \phi_p \circ \psi_p(x).$$

(1.11)

In (1.11), for every $p \in \{1, \ldots, P\}$, $\lambda_p > 0$, $\psi_p : \mathbb{R}^N \to [0, +\infty]$ is proper, lower semi-continuous and Lipschitz continuous on its domain, and $\phi_p : [0, +\infty] \to ]-\infty, +\infty[\text{ is a concave, strictly increasing and differentiable function, such that } \phi_p \circ \psi_p \text{ is Lipschitz-continuous on the domain of } \psi_p$, where $\phi_p$ denotes the first derivative of $\phi_p$. 


To solve problem (1.1), we propose to use a VMFB-based algorithm. Basically, as in (5.2), the Lipschitz differentiable function \( h \) is handled through a gradient step, while the non-smooth term \( g \) is handled using a proximal step. However, due to the composite form of \( g \) given by (1.11), the proximity operator of \( g \) might not be computable, either efficiently, or at all. To overcome this difficulty, we propose to replace, at each iteration \( k \in \mathbb{N} \), the function \( g \) by an approximation, denoted by \( q(\cdot, x_k) : \mathbb{R}^N \to ]-\infty, +\infty] \). Precisely, this approximation is chosen to be a majorant function of \( g \) at \( x_k \):

\[
\forall x \in \mathbb{R}^N \quad \begin{cases} g(x) \leq q(x, x_k), \\ g(x_k) = q(x_k, x_k). \end{cases} \tag{1.12}
\]

Since function \( g \) is the sum of \( P \) composite functions, we choose

\[
\forall x \in \mathbb{R}^N \quad q(x, x_k) = \sum_{p=1}^{P} q_p(x, x_k), \tag{1.13}
\]

such that, for every \( p \in \{1, \ldots, P\} \), \( q_p(\cdot, x_k) : \mathbb{R}^N \to ]-\infty, +\infty] \) is a majorant function of \( \lambda_p \phi_p \circ \psi_p \) at \( x_k \), i.e.

\[
\forall x \in \mathbb{R}^N \quad \begin{cases} \lambda_p \phi_p \circ \psi_p(x) \leq q_p(x, x_k), \\ \lambda_p \phi_p \circ \psi_p(x_k) = q_p(x_k, x_k). \end{cases} \tag{1.14}
\]

and is obtained by taking, for every \( p \in \{1, \ldots, P\} \), the tangent of the concave differentiable function \( \phi_p \) at \( \psi_p(x_k) \):

\[
\forall x \in \mathbb{R}^N \quad q_p(x, x_k) = \lambda_p \phi_p \circ \psi_p(x_k) + \lambda_p \phi_p' \circ \psi_p(x_k) (\psi_p(x) - \psi_p(x_k)). \tag{1.15}
\]

Then, at each iteration \( k \in \mathbb{N} \), the proposed VMFB algorithm with approximated proximal step reads

\[
x_{k+1} = \text{prox}_{q(\cdot, x_k)}(x_k - \gamma_k A_k^{-1} \nabla h(x_k)), \tag{1.16}
\]

where \( \gamma_k > 0 \), and \( A_k \in \mathbb{R}^{N \times N} \) is an SPD matrix. In addition, in order to avoid to compute the approximated function \( q(\cdot, x_k) \) at each iteration, we propose to fix it for a given finite number of iterations. More precisely, at each iteration \( k \in \mathbb{N} \), the majorant function \( q(\cdot, x_k) \) is computed using the current iterate \( x_k \), and kept fixed for \( I_k \in \mathbb{N}^* \) VMFB iterations. Then, the proposed variable metric composite function forward-backward (C2FB) algorithm to solve problem (1.1)-(1.11) is given by

\[
\begin{align*}
x_0 & \in \text{dom } g, \\
\text{for } k = 0, 1, \ldots, \\
\tilde{x}_{k,0} & = x_k, \\
\text{for } i = 0, \ldots, I_k - 1, \\
\tilde{x}_{k,i+1} & = \text{prox}_{q(\cdot, x_k)}(\tilde{x}_{k,i} - \gamma_k A_k^{-1} \nabla h(\tilde{x}_{k,i})), \\
x_{k+1} & = \tilde{x}_{k,I_k}. \tag{1.17}
\end{align*}
\]
where, for every $k \in \mathbb{N}$, and every $i \in \{0, \ldots, I_k - 1\}$, $\gamma_{k,i} > 0$, and $A_{k,i} \in \mathbb{R}^{N \times N}$ is an SPD matrix.

One can notice that in the case when $I_k \to \infty$ for every $k \in \mathbb{N}$, algorithm (1.17) can be interpreted as an MM algorithm of the same flavour as those proposed in [23, 26, 37, 38]. Indeed, in this case, at each iteration $k \in \mathbb{N}$, we have

$$x_{k+1} \approx \arg\min_{x \in \mathbb{R}^N} h(x) + q(x, x_k).$$

(1.18)

In this work, we prove the convergence of sequences $(x_k)_{k \in \mathbb{N}}$ generated by the C2FB algorithm given in (1.17) to a critical point of $f$, using the KL inequality. Subsequently, according to the remark above, we show that algorithm (1.18) converges to a critical point of $f$, if each sub-problem is solved using VMFB iterates, independently of the number of iterations towards solving each sub-problem.

The remainder of the paper is organized as follows. We introduce our notation and give useful definitions of non-convex optimization in Section 2. The proposed C2FB method, including an inexact version allowing the proximity operator to be computed inexactly, are given in Section 3. In Section 4, we investigate the asymptotic behaviour of the proposed method, and we give the main convergence result of this work. Particular cases of the proposed approach, including reweighting algorithms, are described in Section 5. Finally, simulation results on a small image restoration problem are provided in Section 6.

2 Optimization background

In this section we give the definitions and notation used throughout the paper. For additional definitions on non-convex optimization, we refer the reader, e.g., to [44].

**Definition 2.1** Let $f : \mathbb{R}^N \to ]-\infty, +\infty]$.

(i) The level set of $f$ at height $\delta \in \mathbb{R}$ is defined as $\text{lev}_\leq\delta f = \{x \in \mathbb{R}^N \mid f(x) \leq \delta\}$.

(ii) The domain of $f$ is defined as $\text{dom} f = \{x \in \mathbb{R}^N \mid f(x) < +\infty\}$.

(iii) The function $f$ is proper if its domain is non-empty.

**Definition 2.2** Let $f : \mathbb{R}^N \to ]-\infty, +\infty]$ and $\bar{x} \in \mathbb{R}^N$. The Fréchet sub-differential of $f$ at $\bar{x}$ is denoted by $\hat{\partial}f(\bar{x})$, and is given by

$$\hat{\partial}f(\bar{x}) = \left\{ \hat{v}(\bar{x}) \in \mathbb{R}^N \mid \liminf_{y \to \bar{x}, y \neq \bar{x}} \frac{1}{\|y - \bar{x}\|} (f(y) - f(\bar{x}) - \langle y - \bar{x} \mid \hat{v}(\bar{x}) \rangle) \geq 0 \right\}.$$  

(2.1)

If $\bar{x} \notin \text{dom} f$, then $\hat{\partial}f(\bar{x}) = \emptyset$. 

6
The sub-differential of \( f \) at \( \bar{x} \) is denoted by \( \partial f(\bar{x}) \), and is given by

\[
\partial f(\bar{x}) = \left\{ v(\bar{x}) \in \mathbb{R}^{N} \mid \exists (x_{k}, \hat{v}(x_{k})) \rightarrow (\bar{x}, v(\bar{x})) \right. \\
\text{such that } f(x_{k}) \rightarrow f(\bar{x}) \text{ and } \left( \forall k \in \mathbb{N} \right) \hat{v}(x_{k}) \in \hat{\partial}f(x_{k}) \right\}. \tag{2.2}
\]

**Remark 2.3** An equivalent definition for the Fréchet sub-differential of \( f \) at \( \bar{x} \) is given by [44, Def. 8.3]

\[
\hat{\partial}f(\bar{x}) = \left\{ \hat{v}(\bar{x}) \in \mathbb{R}^{N} \mid (\forall x \in \mathbb{R}^{N}) f(x) \geq f(\bar{x}) + \langle \hat{v}(\bar{x}) \mid x - \bar{x} \rangle + o(|x - \bar{x}|) \right\}. \tag{2.3}
\]

**Remark 2.4** A necessary condition for \( x^\star \in \mathbb{R}^{N} \) to be a minimizer of \( f \) is that \( x^\star \) is a critical point of \( f \), i.e. \( 0 \in \partial f(x^\star) \). If \( f \) is convex, this condition is also sufficient.

**Definition 2.5** Let \( A_{1} \in \mathbb{R}^{N \times N} \) and \( A_{2} \in \mathbb{R}^{N \times N} \) be two SPD matrices. The Loewner partial ordering on \( \mathbb{R}^{N \times N} \) is defined as

\[
A_{1} \succeq A_{2} \iff (\forall x \in \mathbb{R}^{N}) x^{\top}A_{1}x \geq x^{\top}A_{2}x. \tag{2.4}
\]

The weighted norm associated with \( A_{1} \) is defined as, for every \( x \in \mathbb{R}^{N} \),

\[
\|x\|_{A_{1}} = \left( x^{\top}A_{1}x \right)^{1/2}. \tag{2.5}
\]

**Definition 2.6** Let \( f : \mathbb{R}^{N} \rightarrow ]-\infty, +\infty[ \) be a proper, lower-semicontinuous function. Let \( A \in \mathbb{R}^{N \times N} \) be an SPD matrix, and let \( \bar{x} \in \mathbb{R}^{N} \). The proximity operator \( \text{prox}^{A}_{f} : \mathbb{R}^{N} \rightrightarrows \mathbb{R}^{N} \) at \( \bar{x} \) of \( f \) relative to the metric induced by \( A \) is given by

\[
\text{prox}^{A}_{f}(\bar{x}) = \text{Argmin}_{x \in \mathbb{R}^{N}} f(x) + \frac{1}{2}\|x - \bar{x}\|_{A}^{2}. \tag{2.5}
\]

**Remark 2.7**

(i) In the definition of the proximity operator, since \( \| \cdot \|_{A}^{2} \) is coercive and \( f \) is proper and lower-semicontinuous, if \( f \) is bounded from below by an affine function, then, for every \( \bar{x} \in \mathbb{R}^{N} \), \( \text{prox}^{A}_{f}(\bar{x}) \) is a non-empty set.

(ii) If \( f \) is convex, then, for every \( \bar{x} \in \mathbb{R}^{N} \), \( \text{prox}^{A}_{f}(\bar{x}) \) is unique. In addition, if \( A = I_{N} \), then \( \text{prox}^{I_{N}}_{f} \equiv \text{prox}_{f} \) is the proximity operator originally defined in [36].

### 3 Proposed optimization method and assumptions

Before giving the assumptions necessary to prove the convergence of the proposed method, we would emphasize that the C2FB algorithm described in (1.17) can be rewritten using the proximity operator of \( \sum_{p=1}^{P} \psi_{p} \) instead of the proximity operator of \( q(., x_{k}) \).
**Remark 3.1** Let, for every $x \in \mathbb{R}^N$ and $k \in \mathbb{N}$,

$$l_k(x) = \sum_{p=1}^{P} \lambda_{p,k} \psi_p(x) \quad \text{where} \quad (\forall p \in \{1, \ldots, P\}) \quad \lambda_{p,k} = \lambda_p \phi_p \circ \psi_p(x_k). \tag{3.1}$$

Then, by noticing that, for every $k \in \mathbb{N}$, $q(\cdot, x_k) = l_k + \eta_k$, where $\eta_k \in \mathbb{R}$, algorithm (1.17) can be reformulated as follows:

$$x_0 \in \text{dom } g,$$

for $k = 0, 1, \ldots$

$$\begin{align*}
\tilde{x}_{k,0} &= x_k, \\
\text{for } i = 0, \ldots, I_k - 1 \\
\tilde{x}_{k,i+1} &= \text{prox}_{l_k} \left( x_{k,i} - \gamma_{k,i} A_{k,i}^{-1} \nabla h(x_{k,i}) \right), \\
x_{k+1} &= \tilde{x}_{k,I_k}. \tag{3.2}
\end{align*}$$

Using the definition of the proximity operator, we can deduce that, for every $k \in \mathbb{N}$ and $i \in \{0, \ldots, I_k\}$ we have $\tilde{x}_{k,i} \in \cap_{p=1}^{P} \text{dom } \psi_p = \text{dom } g$.

We can observe two particular cases of algorithm (3.2). On the one hand, in the particular case when, for every $k \in \mathbb{N}$, $I_k = 1$, then algorithm (3.2) reads

$$x_0 \in \text{dom } g,$$

for $k = 0, 1, \ldots$

$$x_{k+1} = \text{prox}_{l_k} \left( x_k - \gamma_{k} A_{k}^{-1} \nabla h(x_k) \right), \tag{3.3}$$

where $l_k$ is given by (3.1). Algorithm (3.3) requires to redefine the majorant function $q(\cdot, x_k)$ at each iteration $k \in \mathbb{N}$, while in algorithm (3.2) the majorant function is fixed for a finite number of iterations $I_k$. On the other hand, as emphasized in the introduction, under technical assumptions, in the limit case when, for every $k \in \mathbb{N}$, $I_k \to \infty$, according to [15] the sequence $(\tilde{x}_{k,i})_{i \in \mathbb{N}}$ converges to a critical point of $h + l_k$. In other words, each inner-loop in algorithm (3.2) corresponds to the VMFB algorithm as defined in [15], for minimizing $h + l_k$.

### 3.1 Assumptions

In the remainder of this work, we will focus on functions $h$ and $g$ satisfying the following assumptions. Examples of functions satisfying the needed assumptions are described in Section 5.

**Assumption 3.2**

\(\text{(i)}\) The function $h: \mathbb{R}^N \to \mathbb{R}$ is differentiable, and has a $\mu$-Lipschitzian gradient, with $\mu > 0$, i.e., for every $(x, x') \in (\mathbb{R}^N)^2$, $\|\nabla h(x) - \nabla h(x')\| \leq \mu \|x - x'\|$. 

8
(ii) For every \( p \in \{1, \ldots, P\} \), \( \psi_p : \mathbb{R}^N \rightarrow [0, +\infty] \) is a lower-semicontinuous and proper function. Moreover, it is Lipschitz-continuous on its domain.

(iii) For every \( p \in \{1, \ldots, P\} \), the function \( \phi_p : [0, +\infty] \rightarrow ]-\infty, +\infty] \) is a concave, strictly increasing and differentiable function (i.e. \( \phi_p(u) > 0 \) for every \( u \in [0, +\infty] \)).

(iv) For every \( p \in \{1, \ldots, P\} \), the function \( \phi_p \circ \psi_p \) is continuous on its domain.

(v) The function \( f \) is coercive, i.e. \( \lim_{\|x\| \rightarrow +\infty} f(x) = +\infty \), and lower bounded.

**Remark 3.3**

(i) According to Assumptions 3.2(i)-(iii), \( f \) is continuous on \( \text{dom } g = \cap_{p=1}^{P} \text{dom } \psi_p \). In addition, according to Assumption 3.2(v), we can deduce that, for every \( x \in \text{dom } g \), \( \text{lev}_{f(x)} f \) is compact ([17, Prop. 11.12]).

(ii) According to Assumption 3.2(ii), there exists \( \nu > 0 \) such that, for every \( p \in \{1, \ldots, P\} \), \( \|r_p(x)\| \leq \nu \) for every \( r_p(x) \in \partial \psi_p(x) \), with \( x \in \text{dom } \psi_p \). Note that this assumption is satisfied for simple choices of \( \psi_p \) (e.g. \( \psi_p = \|\cdot\|_1 \)), and see Section 5 for more examples).

(iii) According to Assumptions 3.2(ii)-(iii), for every \( k \in \mathbb{N} \) and \( p \in \{1, \ldots, P\} \), the parameter \( \lambda_{p,k} \) introduced in (3.1) is well defined and \( \lambda_{p,k} > 0 \).

For every \( k \in \mathbb{N} \), the SPD matrices \( (A_{k,i})_{0 \leq i \leq I_k - 1} \) are used in practice to accelerate the convergence of usual FB methods. They are chosen using the method proposed in [15, 16, 25], leveraging an MM approach. We define them as follows:

**Assumption 3.4** Let, for every \( k \in \mathbb{N} \), \( (\tilde{x}_{k,i})_{0 \leq i \leq I_k} \) be a sequence generated by algorithm (1.17).

(i) For every \( k \in \mathbb{N} \), we have

\[
(\forall x \in \mathbb{R}^N) \quad h(x) \leq h(\tilde{x}_{k,i}) + \langle x - \tilde{x}_{k,i} \mid \nabla h(\tilde{x}_{k,i}) \rangle + \frac{1}{2}\|x - \tilde{x}_{k,i}\|^2_{A_{k,i}},
\]

(3.4)

(ii) There exists \((\underline{\nu}, \underline{\nu}) \in ]0, +\infty[^2 \) such that, for every \( k \in \mathbb{N} \) and \( i \in \{0, \ldots, I_k - 1\} \), \( \underline{\nu} \mathbf{1}_N \preceq A_{k,i} \preceq \underline{\nu} \mathbf{1}_N \).

**Remark 3.5** According to [15, Lem. 2.1], under Assumption 3.2(i), Assumption 3.4 is trivially satisfied when choosing, for every \( k \in \mathbb{N} \) and for every \( i \in \{0, \ldots, I_k - 1\} \), \( A_{k,i} = \mu \mathbf{1}_N \) and \( \underline{\nu} = \overline{\nu} = \mu \).

The two last assumptions are made to ensure that the step-sizes \((\gamma_{k,i})_{k \in \mathbb{N}, 0 \leq i \leq I_k - 1} \) are bounded, and that the numbers of inner-iterations \((I_k)_{k \in \mathbb{N}} \) are finite.

**Assumption 3.6** There exists \((\underline{\gamma}, \overline{\gamma}) \in ]0, +\infty[^2 \) such that, for every \( k \in \mathbb{N} \) and for every \( i \in \{0, \ldots, I_k - 1\} \) we have \( \underline{\gamma} \leq \gamma_{k,i} \leq 1 - \overline{\gamma} \).

**Assumption 3.7** There exists \( \overline{T} \in \mathbb{N}^* \) such that, for every \( k \in \mathbb{N} \), \( 0 < I_k \leq \overline{T} < +\infty \).
3.2 Inexact algorithm

In general, the proximity operator relative to an arbitrary metric does not have a closed form expression. This is also true for some evolved functions, even when the preconditioning operators are diagonal matrices. For instance when, for every \( p \in \{1, \ldots, P\} \), \( \psi_p \) is a composition between an \( \ell_1 \)-norm and a non-orthogonal matrix (e.g. to promote sparsity in a redundant dictionary), the computation of the proximity operator of \( \psi_p \) is done iteratively [18]. To circumvent this difficulty, we propose an inexact version of the proposed C2FB algorithm given in (3.2):

Let \( \alpha \in (1/2, +\infty], \beta \in [0, +\infty[ \text{ and } x_0 \in \text{dom } g \), for \( k = 0, 1, \ldots \)

\[
\begin{align*}
\tilde{x}_{k,0} &= x_k, \\
\text{for } i = 0, \ldots, I_k - 1, \\
\text{find } \tilde{x}_{k,i+1} \in \mathbb{R}^N \text{ and, for every } p \in \{1, \ldots, P\}, \\
& \quad \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i+1}) + \langle \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \mid \nabla h(\tilde{x}_{k,i}) \rangle \\
& \quad + \alpha \| \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \|_{A_{k,i}} \leq \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i}), \\
\| \nabla h(\tilde{x}_{k,i}) + \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i+1}) \| & \leq \beta \| \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \|_{A_{k,i}}, \\
x_{k+1} &= \tilde{x}_{k,I_k}.
\end{align*}
\]

(3.5)

In algorithm (3.5), for every \( k \in \mathbb{N} \) and \( i \in \{1, \ldots, I_k - 1\} \), \( A_{k,i} \in \mathbb{R}^{N \times N} \) is an SDP matrix satisfying Assumption 3.4.

Under our assumptions, algorithm (3.5) can be viewed as an inexact version of algorithm (3.2) (or equivalently algorithm (1.17)), where, at each iteration \( k \in \mathbb{N} \), the proximity operator of \( l_k \) (or equivalently \( \sum_p \psi_p \)) can be computed inexactly (i.e. using sub-iterations). This inexact version is common for FB algorithms in a nonconvex context (see e.g. [2, 15, 16]). The first inequality in algorithm (3.5) is called sufficient-decrease condition, while the second inequality is the inexact optimality condition. These two conditions allow to handle possible errors arising when the proximity operator is computed approximately, and are often referred to relative errors.

We now show formally that algorithm (3.5) can be viewed as an inexact version of algorithm (3.2). To see this, let \( (\tilde{x}_{k,i})_{k \in \mathbb{N}, 0 \leq i \leq I_k} \) be a sequence generated by algorithm (3.2). Let \( k \in \mathbb{N} \) and \( i \in \{0, \ldots, I_k - 1\} \). Due to the definition of the proximity operator, we have

\[
l_k(\tilde{x}_{k,i+1}) + \langle \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \mid \nabla h(\tilde{x}_{k,i}) \rangle + \frac{1}{2\gamma_{k,i}} \| \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \|^2_{A_{k,i}} \leq l_k(\tilde{x}_{k,i}).
\]

(3.6)

Noticing that \( l_k = \sum_{p=1}^{P} \lambda_{p,k} \psi_p \) under Assumption 3.6 the first condition in algorithm (3.5) (i.e. sufficient-decrease condition) is obtained with \( \alpha = (1 - \gamma)^{-1}/2 \).
The second condition in algorithm (3.5) (i.e. inexact optimality condition) is obtained combining algorithm (3.2) with Assumptions 3.4(ii) and 3.6. Indeed, using the variational characterization of the proximity operator, we have, for every $k \in \mathbb{N}$ and $i \in \{0, \ldots, I_k - 1\}$,

$$
\tilde{x}_{k,i+1} = \text{prox}_{\gamma_k A_{k,i}} \left( \tilde{x}_{k,i} - \gamma_k A_{k,i}^{-1} \nabla h(\tilde{x}_{k,i}) \right)
\iff \tilde{x}_{k,i} - \gamma_k A_{k,i}^{-1} \nabla h(\tilde{x}_{k,i}) - \tilde{x}_{k,i+1} \in \gamma_k A_{k,i}^{-1} \partial l_k(\tilde{x}_{k,i+1}).
\tag{3.7}
$$

We can note that $\partial l_k = \sum_{p=1}^{P} \lambda_{p,k} \partial \psi_p$. Therefore, for every $p \in \{1, \ldots, P\}$, there exists $r_p(\tilde{x}_{k,i+1}) \in \partial \psi_p(\tilde{x}_{k,i+1})$ such that $\sum_{p=1}^{P} \lambda_{p,k} r_p(\tilde{x}_{k,i+1}) = \gamma_k A_{k,i}^{-1} (\tilde{x}_{k,i} - \tilde{x}_{k,i+1}) - \nabla h(\tilde{x}_{k,i})$. Then, using Assumptions 3.4(ii) and 3.6, we obtain

$$
\|\nabla h(\tilde{x}_{k,i}) + r_p(\tilde{x}_{k,i+1})\| = \gamma_k^{-1} \|A_{k,i} (\tilde{x}_{k,i+1} - \tilde{x}_{k,i})\| 
\leq \gamma_k^{-1} \sqrt{\nu} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\| A_{k,i},
\tag{3.8}
$$

and the inexact optimality condition is obtained with $\beta = \gamma_k^{-1} \sqrt{\nu}$.

4 Convergence analysis

4.1 Descent properties

Lemma 4.1 Let $(x_k)_{k \in \mathbb{N}}$ and $(\tilde{x}_{k,i})_{k \in \mathbb{N}, 0 \leq i \leq I_k}$ be sequences generated by algorithm (3.5). Let $(i_1, i_2) \in \{0, \ldots, I_k - 1\}^2$ be such that $i_1 \leq i_2$. Under Assumptions 3.2, 3.4, 3.6 and 3.7, for every $k \in \mathbb{N}$, we have

$$
h(\tilde{x}_{k,i_2+1}) + \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i_2+1}) 
\leq h(\tilde{x}_{k,i_1}) + \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i_1}) - \alpha \sum_{i=i_1}^{i_2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|^2
\tag{4.1}
$$

where $\alpha > 0$.

Proof. Let $(x_k)_{k \in \mathbb{N}}$ and $(\tilde{x}_{k,i})_{k \in \mathbb{N}, 0 \leq i \leq I_k}$ be sequences generated by algorithm (3.5). Using the sufficient decrease condition in algorithm (3.5) (i.e. the first inequality), we have, for every $k \in \mathbb{N}$ and $i \in \{0, \ldots, I_k - 1\}$,

$$
\sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p(\tilde{x}_{k,i+1}) - \psi_p(\tilde{x}_{k,i}) \right) + \alpha \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|^2 A_{k,i}
\leq - (\tilde{x}_{k,i+1} - \tilde{x}_{k,i} \mid \nabla h(\tilde{x}_{k,i})).
\tag{4.2}
$$
Summing on $i \in \{i_1, \ldots, i_2\}$, we obtain
\[
\sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p(\tilde{x}_{k,i_2+1}) - \psi_p(\tilde{x}_{k,i_1}) \right) + \alpha \sum_{i=i_1}^{i_2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|_{A_{k,i}}^2 \\
\leq - \sum_{i=i_1}^{i_2} (\tilde{x}_{k,i+1} - \tilde{x}_{k,i} | \nabla h(\tilde{x}_{k,i})). \tag{4.3}
\]

According to Assumption 3.4(i), we have
\[
h(\tilde{x}_{k,i+1}) \leq h(\tilde{x}_{k,i}) + \langle \tilde{x}_{k,i+1} - \tilde{x}_{k,i} | \nabla h(\tilde{x}_{k,i}) \rangle + \frac{1}{2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|_{A_{k,i}}^2. \tag{4.4}
\]

Summing the last inequality on $i \in \{i_1, \ldots, i_2\}$, we obtain
\[
- \sum_{i=i_1}^{i_2} (\tilde{x}_{k,i+1} - \tilde{x}_{k,i} | \nabla h(\tilde{x}_{k,i})) \\
\leq h(\tilde{x}_{k,i_1}) - h(\tilde{x}_{k,i_2+1}) + \frac{1}{2} \sum_{i=i_1}^{i_2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|_{A_{k,i}}^2, \tag{4.5}
\]

which, combined with (4.3), leads to
\[
\sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p(\tilde{x}_{k,i_2+1}) - \psi_p(\tilde{x}_{k,i_1}) \right) + \alpha \sum_{i=i_1}^{i_2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|_{A_{k,i}}^2 \\
\leq h(\tilde{x}_{k,i_1}) - h(\tilde{x}_{k,i_2+1}) + \frac{1}{2} \sum_{i=i_1}^{i_2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|_{A_{k,i}}^2, \tag{4.6}
\]

which is equivalent to
\[
h(\tilde{x}_{k,i_2+1}) + \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i_2+1}) \\
\leq h(\tilde{x}_{k,i_1}) + \sum_{p=1}^{P} \lambda_{p,k} \psi_p(\tilde{x}_{k,i_1}) - (\alpha - \frac{1}{2}) \sum_{i=i_1}^{i_2} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|_{A_{k,i}}^2. \tag{4.7}
\]

Then, using Assumption 3.4(ii), there exists $\tilde{\alpha} = \nu(\alpha - 1/2)$ such that (4.1) is satisfied. \(\square\)

**Proposition 4.2** Let $(x_k)_{k \in \mathbb{N}}$ and $(\tilde{x}_{k,i})_{k \in \mathbb{N}, 0 < i < I_k}$ be sequences generated by algorithm (3.5). Under Assumptions 3.2, 3.4, 3.6 and 3.7, for every $k \in \mathbb{N}$ and $i_0 \in \{0, \ldots, I_k - 1\}$, we have
\[
f(\tilde{x}_{k,i_0+1}) \leq f(x_k) - \alpha \sum_{i=0}^{i_0} \|\tilde{x}_{k,i+1} - \tilde{x}_{k,i}\|^2 \tag{4.8}
\]
where $\tilde{\alpha} > 0$ is given in Lemma 4.1.
Proof. Let \((x_k)_{k \in \mathbb{N}}\) and, for every \(k \in \mathbb{N}\), \((\tilde{x}_{k,i})_{0 \leq i \leq I_k}\) be a sequence generated by algorithm (3.5). Let \(i_0 \in \{0, \ldots, I_k - 1\}\). According to (1.14)-(1.15), and using (3.1), we have, for every \(p \in \{1, \ldots, P\}\),

\[
\lambda_p \phi_p \circ \psi_p (\tilde{x}_{k,i_0 + 1}, x_k) \leq q_p (\tilde{x}_{k,i_0 + 1}, x_k) = \lambda_p \phi_p \circ \psi_p (x_k) + \lambda_{p,k} \left( \psi_p (\tilde{x}_{k,i_0 + 1}) - \psi_p (x_k) \right). \tag{4.9}
\]

Summing the last inequality on \(p \in \{1, \ldots, P\}\), we obtain

\[
\sum_{p=1}^{P} \lambda_p \phi_p \circ \psi_p (\tilde{x}_{k,i_0 + 1}) \leq \sum_{p=1}^{P} \lambda_p \phi_p \circ \psi_p (x_k) + \sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p (\tilde{x}_{k,i_0 + 1}) - \psi_p (x_k) \right), \tag{4.10}
\]

which, by definition of \(g\) (see (1.11)), is equivalent to

\[
g(\tilde{x}_{k,i_0 + 1}) \leq g(x_k) + \sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p (\tilde{x}_{k,i_0 + 1}) - \psi_p (x_k) \right). \tag{4.11}
\]

In addition, using Lemma 4.1 with \(i_1 = 0\) and \(i_2 = i_0\), we obtain

\[
\sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p (\tilde{x}_{k,i_0 + 1}) - \psi_p (x_k) \right) \leq h(x_k) - h(\tilde{x}_{k,i_0 + 1}) - \alpha \sum_{i=0}^{i_0} \| \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \|^2. \tag{4.12}
\]

Combining the last two inequalities leads to (4.8).

\[\square\]

Remark 4.3

(i) Let us introduce the sequence \((\chi_k)_{k \in \mathbb{N}}\), defined as, for every \(k \in \mathbb{N}\), \(\chi_k = (\tilde{x}_{k,i+1} - \tilde{x}_{k,i})_{0 \leq i \leq I_k - 1}\).

(ii) For the next proposition, we need to introduce the following additional notation

\[
(\forall x \in \text{dom } g) \quad f_k (x) = h(x) + q(x, x_k). \tag{4.13}
\]

Proposition 4.4 Let \((x_k)_{k \in \mathbb{N}}\) be a sequence generated by algorithm (3.5). Under Assumptions 3.2, 3.4, 3.6 and 3.7, for every \(k \in \mathbb{N}\), we have

\[
f(x_{k+1}) \leq f_k (x_{k+1}) \leq f(x_k) - \bar{\alpha} \| \chi_k \|^2, \tag{4.14}
\]

where \(\bar{\alpha} > 0\) is given in Lemma 4.1.

Proof. The first inequality in (4.14) is obtained directly from equation (1.12).
We now need to show the second inequality. Let \((x_k)_{k \in \mathbb{N}}\) and \((\tilde{x}_{k,i})_{k \in \mathbb{N}, 0 \leq i \leq I_k}\) be sequences generated by algorithm (3.5). According toLemma 4.1, choosing \(i_1 = 0\) and \(i_2 = I_k - 1\), we have

\[
h(x_{k+1}) + \sum_{p=1}^{P} \lambda_{p,k} \left( \psi_p(x_{k+1}) - \psi_p(x_k) \right) \leq h(x_k) - \alpha \sum_{i=0}^{I_k-1} \left\| \tilde{x}_{k,i+1} - \tilde{x}_{k,i} \right\|^2.
\] (4.15)

Then, (4.14) is obtained by noticing that

\[
f_k(x_{k+1}) = h(x_{k+1}) + \sum_{p=1}^{P} \left( \lambda_p \phi_p \circ \psi_p(x_k) + \lambda_{p,k} \left( \psi_p(x_{k+1}) - \psi_p(x_k) \right) \right),
\] (4.16)

and

\[
f(x_k) = h(x_k) + \sum_{p=1}^{P} \lambda_p \phi_p \circ \psi_p(x_k).
\]

\[\square\]

4.2 Convergence results

Before giving our main convergence result, we need to introduce our last assumption, concerning the Kurdyka-Łojasiewicz inequality [4, 5, 6, 29, 33]:

**Assumption 4.5** For every \(k \in \mathbb{N}\), the function \(f_k\) defined by (4.13) satisfies the Kurdyka-Łojasiewicz inequality, i.e., for every \(\xi \in \mathbb{R}\), and, for every bounded subset \(E \subset \mathbb{R}^N\), there exist three constants \(\kappa > 0\), \(\zeta > 0\) and \(\theta \in [0, 1]\) such that

\[
(\forall t_k(x) \in \partial f_k(x)) \quad \|t_k(x)\| \geq \kappa |f_k(x) - \xi|^\theta,
\] (4.17)

for every \(x \in E\) such that \(|f_k(x) - \xi| \leq \zeta\) (using the convention that \(0^0 = 0\)).

**Remark 4.6** As emphasized, e.g., in [1], the KL inequality is satisfied for a wide class of functions, and in particular by real analytic and semi-algebraic functions\(^2\).

According to (1.15) and (4.13), we have, for every \(k \in \mathbb{N}\) and \(x \in \mathbb{R}^N\),

\[
f_k(x) = h(x) + q(x, x_k) = h(x) + \sum_{p=1}^{P} \lambda_p \phi_p(x, x_k)
\]

\[= h(x) + \sum_{p=1}^{P} \lambda_{p,k} \psi_p(x) + C_k,
\] (4.18)

where \(C_k = \sum_{p=1}^{P} \lambda_p \phi_p \circ \psi_p(x_k) - \lambda_p \phi_p \circ \psi_p(x_k) \psi_p(x_k)\) and, for every \(p \in \{1, \ldots, P\}\), \(\lambda_{p,k} > 0\). Since, according to Remark 3.1, \(x_k \in \text{dom } g\), then \(C_k \in \mathbb{R}\). Note that, if \(h\) and, for every \(p \in \{1, \ldots, P\}\), \(\psi_p\) are semi-algebraic functions, then for every \(\bar{\lambda} > 0\), \(h + \bar{\lambda} \sum_p \psi_p\) satisfies the KL inequality. Consequently, if \(h, \psi_1, \ldots, \psi_P\) are semi-algebraic functions, then Assumption 4.5 is satisfied.

\(^2\)A function is semi-algebraic if its graph is a finite union of sets defined by a finite number of polynomial inequalities.
The following theorem, deduced from [44, Prop. 6.6 & Thm. 8.9], will be useful to establish the convergence of the proposed C2FB algorithm to a critical point of $f$.

**Theorem 4.7** Let $(x_k, t(x_k))_{k \in \mathbb{N}}$ be a sequence belonging to $\text{graph} \partial f$. If $(x_k, t(x_k))_{k \in \mathbb{N}}$ converges to $(x^*, t^*)$, and $(f(x_k))_{k \in \mathbb{N}}$ converges to $f(x^*)$, then $(x^*, t^*) \in \text{graph} \partial f$.

The following proposition will be used to link the sub-gradients of the objective function and its majorant.

**Proposition 4.8** Let $\sigma: \mathbb{R}^N \rightarrow [0, +\infty]$ be a proper function which is continuous on its domain, and let $\varphi: [0, +\infty] \rightarrow (-\infty, +\infty]$ be a concave, strictly increasing and differentiable function. We further assume that $\varphi \circ \sigma$ is continuous on its domain. Then, for every $x \in \mathbb{R}^N$, we have $\partial \varphi \circ \sigma(x) = \varphi \circ \sigma(x) \partial \sigma(x)$.

To prove this result, we need to use the following proposition, established in [44, Prop. 8.5].

**Proposition 4.9** (Variational description of regular subgradients) Let $\bar{x} \in \mathbb{R}^N$, $\mathcal{N}(\bar{x})$ be a neighborhood of $\bar{x}$, and $\sigma: \mathbb{R}^N \rightarrow (-\infty, +\infty]$. A vector $v(\bar{x})$ belongs to $\partial \sigma(\bar{x})$ if and only if there exists a function $\rho: \mathbb{R}^N \rightarrow (-\infty, +\infty]$ such that, for every $x \in \mathcal{N}(\bar{x})$, $\rho(x) \leq \sigma(x)$, $\rho(\bar{x}) = \sigma(\bar{x})$, and $\rho$ is differentiable at $\bar{x}$ with $\nabla \rho(\bar{x}) = v(\bar{x})$. Moreover, $\rho$ can be taken to be differentiable with $\rho(x) < \sigma(x)$, for every $x \in \mathcal{N}(\bar{x}) \setminus \{\bar{x}\}$.

We are now ready to prove Proposition 4.8.

**Proof.** Firstly, let us show the first inclusion, i.e., $\partial \varphi \circ \sigma(x) \subset \varphi \circ \sigma(x) \partial \sigma(x)$, for $x \in \mathbb{R}^N$. Let $v(\bar{x}) \in \partial \varphi \circ \sigma(\bar{x})$. According to Definition 2.2, there exists $(x_k, v(x_k))_{k \in \mathbb{N}}$ converging to $(\bar{x}, v(\bar{x}))$, such that $\varphi \circ \sigma(x_k) \rightarrow \varphi \circ \sigma(\bar{x})$ and, for every $k \in \mathbb{N}$, $v(x_k) \in \partial \varphi \circ \sigma(x_k)$. Thus, according to Remark 2.3, for every $x \in \mathbb{R}^N$, we have

$$\varphi \circ \sigma(x) \geq \varphi \circ \sigma(x_k) + \langle v(x_k) | x - x_k \rangle + o(|x - x_k|). \tag{4.19}$$

Since $\varphi$ is a concave and differentiable function, we have

$$\langle u_1, u_2 \rangle \in [0, +\infty]^2 \hspace{1em} \varphi(u_1) - \varphi(u_2) \leq \varphi(u_2)(u_1 - u_2). \tag{4.20}$$

Let $u_1 = \sigma(x)$ and $u_2 = \sigma(x_k)$, then

$$\varphi \circ \sigma(x) - \varphi \circ \sigma(x_k) \leq \varphi \circ \sigma(x_k)(\sigma(x) - \sigma(x_k)). \tag{4.21}$$

Combining the last inequality with (4.19) leads to

$$\varphi \circ \sigma(x_k)(\sigma(x) - \sigma(x_k)) \geq \langle v(x_k) | x - x_k \rangle + o(|x - x_k|). \tag{4.22}$$

Since $\varphi$ is a strictly increasing function, for every $u \in [0, +\infty]$, $\hat{\varphi}(u) > 0$. Then $\varphi \circ \sigma(x_k) \neq 0$, and the last inequality is equivalent to

$$\sigma(x) \geq \sigma(x_k) + \langle (\varphi \circ \sigma(x_k))^{-1}v(x_k) | x - x_k \rangle + o(|x - x_k|). \tag{4.23}$$
Then, by definition of \( \hat{\partial} \sigma(x_k) \), we have \((\varphi \circ \sigma(x_k))^{-1}v(x_k) \in \hat{\partial} \sigma(x_k)\). In addition, since \( x_k \to \bar{\pi} \) and \( \sigma \) is a continuous function, we have \( \sigma(x_k) \to \sigma(\bar{\pi}) \). Finally, since \( v(x_k) \to v(\bar{\pi}) \in \mathbb{R}^n \), \( \varphi \circ \sigma \) is continuous on \( \text{dom} \ (\varphi \circ \sigma) \), and \( \varphi \circ \sigma(x_k) \in ]0, +\infty[ \), we have \((\varphi \circ \sigma(x_k))^{-1}v(x_k) \to (\varphi \circ \sigma(\bar{\pi}))^{-1}v(\bar{\pi})\). Therefore, using Definition 2.2, we can conclude that \((\varphi \circ \sigma(\bar{\pi}))^{-1}v(\bar{\pi}) \in \partial \sigma(\bar{\pi})\), i.e. \( v(\bar{\pi}) \in \varphi \circ \sigma(\bar{\pi}) \partial \sigma(\bar{\pi}) \).

We will now show the second inclusion, i.e. \( \partial \varphi \circ \sigma(\bar{\pi}) \supset \varphi \circ \sigma(\bar{\pi}) \partial \sigma(\bar{\pi}) \), for \( \bar{\pi} \in \mathbb{R}^n \). Let \( v(\bar{\pi}) = \varphi \circ \sigma(\bar{\pi})r(\bar{\pi}) \), with \( r(\bar{\pi}) \in \partial \sigma(\bar{\pi}) \). Then, according to Definition 2.2, there exists a sequence \( (x_k, r(x_k))_{k \in \mathbb{N}} \) converging to \( (\bar{\pi}, r(\bar{\pi})) \), such that \( \sigma(x_k) \to \sigma(\bar{\pi}) \) and \( r(x_k) \in \partial \sigma(x_k) \). According to Proposition 4.9, for every \( k \in \mathbb{N} \), on a neighborhood \( \mathcal{N}(x_k) \) of \( x_k \), there exists a differentiable function \( \rho_k \) such that \( \nabla \rho_k(x_k) = r(x_k) \), \( \rho_k(x_k) = \sigma(x_k) \), and, for every \( x \in \mathcal{N}(x_k) \setminus \{x_k\} \), \( \rho_k(x) < \sigma(x) \). Let \( \bar{\rho}_k = \varphi \circ \rho_k \). Since \( \varphi \) is differentiable, the function \( \bar{\rho}_k \) is also differentiable, and we have

\[
\nabla \bar{\rho}_k(x_k) = \varphi \circ \rho_k(x_k) \nabla \rho_k(x_k) = \varphi \circ \rho_k(x_k) r(x_k) = \varphi \circ \sigma(x_k) r(x_k) \quad (4.24)
\]

In addition, we have \( \bar{\rho}_k(x_k) = \varphi \circ \sigma(x_k) \) and, since \( \varphi \) is a continuous and strictly increasing function, for every \( x \in \mathcal{N}(x_k) \setminus \{x_k\} \), \( \bar{\rho}_k(x) = \varphi \circ \rho_k(x) < \varphi \circ \sigma(x) \). Then, using again Proposition 4.9, we deduce that

\[
v(x_k) := \varphi \circ \sigma(x_k) r(x_k) \in \hat{\partial} \varphi \circ \sigma(x_k) \quad (4.25)
\]

In addition, since \( x_k \to \bar{\pi} \) and \( \varphi \circ \sigma \) is a continuous function, we have

\[
\varphi \circ \sigma(x_k) \to \varphi \circ \sigma(\bar{\pi}) \quad (4.26)
\]

Finally, since \( x_k \to \bar{\pi} \) and \( \varphi \circ \sigma \) is continuous, we have \( \varphi \circ \sigma(x_k) \to \varphi \circ \sigma(\bar{\pi}) \). In addition, since \( r(x_k) \to r(\bar{\pi}) \), we have

\[
v(x_k) \to \varphi \circ \sigma(\bar{\pi}) r(\bar{\pi}) = v(\bar{\pi}) \quad (4.27)
\]

According to Definition 2.2, we can deduce that \( v(\bar{\pi}) \in \partial \varphi \circ \sigma(\bar{\pi}) \). \( \square \)

Finally, the following lemma, particular case of [15, Lem. 3.3] will be useful to prove the main convergence theorem.

**Lemma 4.10** Let \( (u_k)_{k \in \mathbb{N}}, (u'_k)_{k \in \mathbb{N}} \) and \( (\Delta'_k)_{k \in \mathbb{N}} \) be sequences of non-negative reals, and let \( \theta \in ]0, 1[ \). Assume that

(i) For every \( k \in \mathbb{N} \), \( u^2_k \leqslant (u'_k)^\theta \Delta'_k \).

(ii) \( (\Delta'_k)_{k \in \mathbb{N}} \) is summable.

(iii) For every \( k \geqslant k^* \), \( (u'_{k+1})^\theta \leqslant \beta u_k \), where \( \beta > 0 \) and \( k^* \in \mathbb{N} \).

Then, \( (u_k)_{k \in \mathbb{N}} \) is a summable sequence.
The next theorem is our main convergence result, analysing the convergence of the sequences generated by the C2FB algorithm given in (3.5).

**Theorem 4.11** Let \((x_k)_{k \in \mathbb{N}}\) be a sequence generated by algorithm (3.5). Under Assumptions 3.2, 3.4, 3.6, 3.7 and 4.5, the following holds.

(i) \((x_k)_{k \in \mathbb{N}}\) converges to a critical point \(x^*\) of \(f\).

(ii) \((x_k)_{k \in \mathbb{N}}\) is summable, i.e. \(\sum_{k=0}^{+\infty} ||x_{k+1} - x_k|| < +\infty\).

(iii) \((f(x_k))_{k \in \mathbb{N}}\) is a non-increasing sequence converging to \(f(x^*)\).

**Proof.** According to Proposition 4.4, we have, for every \(k \in \mathbb{N}\), \(f(x_{k+1}) \leq f(x_k)\), thus, \((f(x_k))_{k \in \mathbb{N}}\) is a non-increasing sequence. In addition, according to Remark 3.1, for every \(k \in \mathbb{N}\), \(x_k \in \text{dom } g = \text{dom } f\), and using Remark 3.3(i), we can deduce that the sequence \((x_k)_{k \in \mathbb{N}}\) belongs to a compact subset \(E\) of \(\text{lev}_{\leq f(x_0)} f \subset \text{dom } f\). Thus, \(f\) being lower bounded (according to Assumption 3.2(v)), \((f(x_k))_{k \in \mathbb{N}}\) converges to a variable \(\xi \in \mathbb{R}\), and \((f(x_k) - \xi)_{k \in \mathbb{N}}\) is a non-negative sequence converging to 0. According to Proposition 4.4, the sequence \((f_k(x_{k+1}))_{k \in \mathbb{N}}\) is also a non-increasing sequence, converging to \(\xi\).

According to (4.14), we have

\[
\bar{\alpha} \|x_k\|^2 \leq (f(x_k) - \xi) - (f(x_{k+1}) - \xi). \tag{4.28}
\]

Let \(\delta : [0, +\infty] \to [0, +\infty] : t \mapsto t^{1/(1-\theta)}\), where \(\theta \in [0, 1]\). The function \(\delta\) is convex, and we have, for every \((u_1, u_2) \in [0, +\infty)^2\), \(\delta(u_1) - \delta(u_2) \leq \delta(1)(u_1 - u_2)\), with \(\delta(1) = (1 - \theta)^{-1} u_1^{\theta/(1-\theta)}\). By a change of variable, we obtain

\[
u_1 - u_2 \leq (1 - \theta)^{-1} u_1^{\theta} (u_1^{1-\theta} - u_2^{1-\theta}). \tag{4.29}
\]

Then, setting \(u_1 = f(x_k) - \xi\) and \(u_2 = f(x_{k+1}) - \xi\) leads to

\[
\bar{\alpha} \|x_k\|^2 \leq (1 - \theta)^{-1} (f(x_k) - \xi)^{\theta} \Delta_k, \tag{4.30}
\]

where \(\Delta_k = (f(x_k) - \xi)^{1-\theta} - (f(x_{k+1}) - \xi)^{1-\theta}\) is a summable sequence in the sense \(\sum_{k=0}^{+\infty} \Delta_k = (f(x_0) - \xi)^{1-\theta}\).

According to Assumption 4.5, since \(E\) is bounded, there exist \(k > 0\), \(\theta \in [0, 1]\) and \(\zeta > 0\) such that (4.17) holds for every \(x \in E\) for which the inequality \(|f_k(x) - \xi| \leq \zeta\) is satisfied. Since \((f_k(x_{k+1}) - \xi)_{k \in \mathbb{N}}\) converges to 0, there exist \(k^* \in \mathbb{N}\) such that, for every \(k \geq k^*\), \(|f_k(x_{k+1}) - \xi| < \zeta\). Hence, for every \(k \geq k^*\), we have

\[
(\forall t_k(x_{k+1}) \in \partial f_k(x_{k+1})) \quad \kappa |f_k(x_{k+1}) - \xi|^{\theta} \leq ||t_k(x_{k+1})||. \tag{4.31}
\]

By definition of \(f_k\), we have \(t_k(x_{k+1}) \in \nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_{p,k} \partial \psi_p(x_{k+1})\) \(= \nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_{p,k} \partial \psi_p(x_{k+1})\). Let \(r_p(x_{k+1}) \in \partial \psi_p(x_{k+1})\) be defined as in algorithm (3.5).
Then, combining the two last equations, we obtain
\[ \kappa |f_k(x_{k+1}) - \xi|^\theta \leq \| \nabla h(x_{k+1}) + \sum_{p=1}^P \lambda_{p,k} r_p(x_{k+1}) \|. \] (4.32)

Using Jensen’s inequality, we have
\[
\| \nabla h(x_{k+1}) + \sum_{p=1}^P \lambda_{p,k} r_p(x_{k+1}) \|^2 \\
= \| \nabla h(\bar{x}_{k,I_k}) + \sum_{p=1}^P \lambda_{p,k} r_p(\bar{x}_{k,I_k}) \|^2 \\
\leq \sum_{i=0}^{I_k-1} \| \nabla h(\bar{x}_{k,i+1}) + \sum_{p=1}^P \lambda_{p,k} r_p(\bar{x}_{k,i+1}) \|^2 \\
\leq 2 \sum_{i=0}^{I_k-1} \left( \| \nabla h(\bar{x}_{k,i+1}) - \nabla h(\bar{x}_{k,i}) \|^2 + \| \nabla h(\bar{x}_{k,i}) + \sum_{p=1}^P \lambda_{p,k} r_p(\bar{x}_{k,i+1}) \|^2 \right) \\
\leq 2 \sum_{i=0}^{I_k-1} \left( \mu^2 \| \bar{x}_{k,i+1} - \bar{x}_{k,i} \|^2 + \nu \beta^2 \| \bar{x}_{k,i+1} - \bar{x}_{k,i} \|^2 \right),
\] (4.33)
where the last majoration is obtained using Assumption 3.2(i), Assumption 3.2(ii) and the second inequality condition in algorithm (3.5). Then, by definition of \((\chi_k)_{k \in \mathbb{N}}\),
\[ \| \nabla h(x_{k+1}) + \sum_{p=1}^P \lambda_{p,k} r_p(x_{k+1}) \| \leq \sqrt{2(\mu^2 + \nu \beta)} \| \chi_k \|. \] (4.34)

Therefore, by combining equations (4.32) and (4.34), and using Proposition 4.4, we obtain
\[ |f(x_{k+1}) - \xi|^\theta \leq |f_k(x_{k+1}) - \xi|^\theta \leq \kappa^{-1} \sqrt{2(\mu^2 + \nu \beta)} \| \chi_k \|. \] (4.35)

Applying Proposition 4.10 for \( \theta \in ]0, 1[, \) with \( u_k = \| \chi_k \|, u'_k = |f(x_k) - \xi|, \Delta'_k = (\pi(1 - \theta))^{-1} \Delta_k \) and \( \beta = \kappa^{-1} \sqrt{2(\mu^2 + \nu \beta)} \), we conclude that \((\| \chi_k \|)_{k \in \mathbb{N}}\) is summable.

It remains to show that \((\| \chi_k \|)_{k \in \mathbb{N}}\) is summable when \( \theta = 0 \). According to (4.28), we can deduce that the sequence \((\chi_k)_{k \in \mathbb{N}}\) converges to 0. Thus, there exists \( k^{**} \geq k^* \) such that, for every \( k \geq k^{**}, \) \( \kappa^{-1} \sqrt{2(\mu^2 + \nu \beta)} \| \chi_k \| < 1 \). Hence, according to (4.35), using the convention \( 0^0 = 0 \), we have, for every \( k \geq k^{**}, \) \( f(x_{k+1}) = \xi \). Therefore, according to (4.30), for every \( k \geq k^{**}, \chi_k = 0 \), which trivially shows that \((\| \chi_k \|)_{k \in \mathbb{N}}\) is a summable sequence.

According to Jensen’s inequality and by definition of \((\chi_k)_{k \in \mathbb{N}}\), for every \( k \in \mathbb{N} \) we have
\[ \| x_{k+1} - x_k \|^2 \leq I_k \sum_{i=0}^{I_k-1} \| \bar{x}_{k,i+1} - \bar{x}_{k,i} \|^2 = I_k \| \chi_k \|^2. \] Then, according to Assumption 3.7, we have \( \| x_{k+1} - x_k \| \leq \sqrt{7} \| \chi_k \|, \) and thus \((\| x_{k+1} - x_k \|)_{k \in \mathbb{N}}\) is a summable sequence. Subsequently, we deduce that \((x_k)_{k \in \mathbb{N}}\) is a Cauchy sequence, hence converging to a point \( x^* \). In addition, according to Remark 3.3(i), \( f \) being a continuous function, the sequence \((f(x_k))_{k \in \mathbb{N}}\) converges to \( f(x^*) \).
It remains to show that $x^*$ is a critical point of $f$. Let $t(x_{k+1}) \in \partial f(x_{k+1})$. By definition of $f$, using Proposition 4.8 and using the fact that, for every $p \in \{1, \ldots, P\}$, $\lambda_{p,k+1} = \lambda_p \phi_p \circ \psi_p(x_{k+1})$, we have

$$
t(x_{k+1}) \in \partial f(x_{k+1}) = \nabla h(x_{k+1}) + \partial g(x_{k+1})$$

$$= \nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_p \partial \phi_p \circ \psi_p(x_{k+1})$$

$$= \nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_{p,k+1} \partial \psi_p(x_{k+1}).$$

(4.36)

Let, for every $p \in \{1, \ldots, P\}$, $r_p(x_{k+1}) \in \partial \psi_p(x_{k+1})$ given in equation (4.32), then, using Remark 3.3(ii), we have

$$||t(x_{k+1})|| = ||\nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_{p,k+1} r_p(x_{k+1})||$$

$$\leq ||\nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_p \partial \phi_p(x_{k+1})|| + \sum_{p=1}^{P} (\lambda_{p,k+1} - \lambda_p) r_p(x_{k+1})||$$

$$\leq ||\nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_p \partial \phi_p(x_{k+1})|| + \sum_{p=1}^{P} ||\lambda_{p,k+1} - \lambda_p|| ||r_p(x_{k+1})||$$

$$\leq ||\nabla h(x_{k+1}) + \sum_{p=1}^{P} \lambda_p \partial \phi_p(x_{k+1})|| + ||\lambda_{p,k+1} - \lambda_p||.$$

(4.37)

Using (4.34), we have $||t(x_{k+1})|| \leq \sqrt{2(\mu^2 + \beta^2)} ||\chi_k|| + \sum_{p=1}^{P} \nu ||\lambda_{p,k+1} - \lambda_p||$. Noticing that, for every $p \in \{1, \ldots, P\}$, $\lambda_{p,k+1} - \lambda_p = \lambda_p (\phi_p \circ \psi_p(x_{k+1}) - \phi_p \circ \psi_p(x_k))$, since $\phi_p \circ \psi_p$ is continuous, we have $\lambda_{p,k+1} - \lambda_p \to 0$. In addition, $(\chi_k)_{k \in \mathbb{N}}$ is summable, thus it converges to 0. Hence we deduce that $(x_k, t(x_k))_{k \in \mathbb{N}}$ converges to $(x^*, 0)$. Therefore, using Theorem 4.7, we conclude that $(x^*, 0) \in \text{graph} \partial f$, hence $x^*$ is a critical point of $f$.

\[ \Box \]

5 Particular cases of the proposed method

In this section we describe particular cases for the proposed C2FB algorithm, paying attention to the assumptions necessary to ensure that the convergence of Theorem 4.11 holds.

5.1 Variable Metric Forward Backward algorithm

The proposed algorithm boils down to the VMFB developed in [15] when choosing $P = 1$, $\lambda_1 = 1$, $\phi_1 = \text{Id}$, and $\psi_1 = \psi$ being a proper and Lipschitz-continuous function. In this case, we have $g = \psi$.
and the global minimization problem is of the form
\[
\minimize_{x \in \mathbb{R}^N} h(x) + \psi(x).
\] (5.1)

In this particular case, we have, for every \( x \in \mathbb{R}^N \), \( q(x, x_k) = \psi(x) \), and consequently, algorithms (1.17) reduces to
\[
\begin{align*}
x_0 & \in \text{dom } g, \\
\text{for } k = 0, 1, \ldots \rightarrow \\
x_{k+1} &= \text{prox}_{\gamma_k \psi} A_k(x_k - \gamma_k A_k^{-1} \nabla h(x_k)).
\end{align*}
\] (5.2)

It is important to emphasize here that the assumptions on \( \psi \) to ensure convergence of (5.2) are different than in [15]. On the one hand, in [15] the function \( \psi \) only needs to be continuous on its domain (while in Assumption (ii) we assume that \( \psi \) is Lipschitz-continuous on its domain), but necessitates to be convex. Other works present convergence results for algorithm (5.2) when \( \psi \) is non-convex either in the case without variable metric [2] or for alternating minimization [16]. On the other hand, in [15] the step-sizes \((\gamma_k)_{k \in \mathbb{N}}\) are chosen such that, there exists \((\underline{\gamma}, \overline{\gamma}) \in ]0, +\infty[^2\) such that, for every \( k \in \mathbb{N}, \underline{\gamma} \leq \gamma_k \leq 2 - \overline{\gamma} \), while in Assumption 3.6 it is assumed that \( \underline{\gamma} \leq \gamma_k \leq 1 - \overline{\gamma} \).

5.2 Reweighted algorithms

The second example is interesting in computational imaging since it is related to iteratively reweighted algorithms [37].

We consider problem (1.1)-(1.11) with, for every \( p \in \{1, \ldots, P\} \), \( \psi_p \) and \( \lambda_p \) satisfying Assumption 3.2 and, for every \( u \in ]0, +\infty[ \), \( \phi_p(u) = \log(u + \varepsilon) \), where \( \varepsilon > 0 \). In this case, we have
\[
(\forall x \in \mathbb{R}^N) \quad g(x) = \sum_{p=1}^{P} \lambda_p \log(\psi_p(x) + \varepsilon),
\] (5.3)
and, for every \( k \in \mathbb{N} \) and \( p \in \{1, \ldots, P\} \),
\[
(\forall x \in \mathbb{R}^N) \quad q_p(x, x_k) = \lambda_p \log(\psi_p(x_k) + \varepsilon) + \lambda_{p,k}(\psi_p(x) - \psi_p(x_k)),
\] (5.4)
with \( \lambda_{p,k} = \lambda_p(\psi_p(x_k) + \varepsilon)^{-1} \).
In this context, the proposed algorithm given in (3.2) reduces to

\[ x_0 \in \text{dom } g, \]
for \( k = 0, 1, \ldots \)
\[ \begin{aligned}
  \text{for } p = 1, \ldots, P \\
  \quad \lambda_{p,k} = \lambda_p(\psi_p(x_k) + \varepsilon)_k^{-1}, \\
  \quad \tilde{x}_{k,0} = x_k, \\
  \text{for } i = 0, \ldots, I_k - 1 \\
  \quad \tilde{x}_{k,i+1} = \text{prox}_{\gamma_k^{-1}A_k,i}(\tilde{x}_{k,i} - \gamma_k,iA_k^{-1}\nabla h(\tilde{x}_{k,i})), \\
  \quad x_{k+1} = \tilde{x}_{k,I_k}.
\end{aligned} \] (5.5)

5.2.1 Reweighted \( \ell_1 \) algorithm for log-sum penalization

In the particular case where, for every \( p \in \{1, \ldots, P\} \),

\[ (\forall x \in \mathbb{R}^N) \quad \psi_p(x) = ||Wx||^p, \] (5.6)

with \( W : \mathbb{R}^N \to \mathbb{R}^P \) is a linear operator (e.g. wavelet transform [35]), then we have, for every \( k \in \mathbb{N}, \lambda_{p,k} = \lambda_p(||Wx||^p + \varepsilon)_k^{-1}. \) In this context, the function \( g \) corresponds to a log-sum penalization composed with a linear operator:

\[ (\forall x \in \mathbb{R}^N) \quad g(x) = \sum_{p=1}^P \lambda_p \log (||Wx||^p + \varepsilon), \] (5.7)

and algorithm (5.5) reduces to the re-weighted \( \ell_1 \) algorithm, initially proposed in [11] for \( W = I_N \), where each sub-problem is solved using a VMFB algorithm. The resulting algorithm reads

\[ x_0 \in \text{dom } g, \]
for \( k = 0, 1, \ldots \)
\[ \begin{aligned}
  \Lambda_k &= \text{Diag} \left( \left( \frac{\lambda_p}{||Wx||^p + \varepsilon}_k \right)_{1 \leq p \leq P} \right), \\
  \tilde{x}_{k,0} &= x_k, \\
  \text{for } i = 0, \ldots, I_k - 1 \\
  \quad \tilde{x}_{k,i+1} = \text{prox}_{\gamma_k^{-1}A_k,i}\|\Lambda_k W^{-1}\|_1(\tilde{x}_{k,i} - \gamma_k,iA_k^{-1}\nabla h(\tilde{x}_{k,i})), \\
  \quad x_{k+1} = \tilde{x}_{k,I_k}.
\end{aligned} \] (5.8)

where \( \text{Diag}(\cdot) \) is the operator giving the diagonal matrix whose diagonal elements are given by its argument.

Note that (5.7) is also known as the log-sum penalization and its proximity operator has an explicit form [14]. Let \( A = \text{Diag} \left( (a^{(n)})_{1 \leq n \leq N} \right) \in [0, +\infty]^{N \times N} \) be a diagonal SDP matrix. Then the
proximity operator relatively to the metric induced by $A$ is given by:

$$(\forall x \in \mathbb{R}^N) \quad \text{prox}_A^g(x) = (\mathcal{P}^{(n)})_{1 \leq n \leq N} \quad \text{where} \quad (\forall n \in \{1, \ldots, N\})$$

$$\mathcal{P}^{(n)} = \begin{cases} 0, & \text{if } |x^{(n)}| < \sqrt{4a^{(n)} - \varepsilon}, \\ \max \left\{ 0, \frac{\text{sign}(x^{(n)}) \sqrt{|x^{(n)}| - \varepsilon + \sqrt{(|x^{(n)}| + \varepsilon)^2 - 4a^{(n)}}}}{2}, \frac{\text{sign}(x^{(n)}) \sqrt{|x^{(n)}| - \varepsilon + \sqrt{(|x^{(n)}| + \varepsilon)^2 - 4a^{(n)}}}}{2} \right\}, & \text{if } |x^{(n)}| = \sqrt{4a^{(n)} - \varepsilon}, \\ \text{sign}(x^{(n)}) \frac{|x^{(n)}| - \varepsilon + \sqrt{(|x^{(n)}| + \varepsilon)^2 - 4a^{(n)}}}{2} + \sqrt{|x^{(n)}| + \varepsilon) - \varepsilon}, & \text{otherwise}. \end{cases} (5.9)$$

Consequently, one can apply the VMFB algorithm, for diagonal matrices $(A_k)_{k \in \mathbb{N}}$, directly to minimize $f = h + g$, when $h$ is a Lipschitz differentiable function and $g$ is defined in (5.7).

To emphasize the advantage of the proposed approach, we provide an illustration in the context of imaging inverse problem comparing algorithm (5.8) and the VMFB algorithm directly applied to minimize $f = h + g$ (see Section 6.1).

5.2.2 Cauchy penalization

Similarly to the reweighting $\ell_1$ algorithm described above, another particular case is when choosing $P = N$ and, for every $n \in \{1, \ldots, N\}$ and $x \in \mathbb{R}^N$, $\psi_n(x) = (x^{(n)})^2$. In this case, when choosing for every $n \in \{1, \ldots, N\}$, $\lambda_n \equiv \lambda > 0$, $g$ corresponds to the Cauchy penalization of the form

$$(\forall x \in \mathbb{R}^N) \quad g(x) = \lambda \sum_{n=1}^{N} \log \left( (x^{(n)})^2 + \varepsilon \right), (5.10)$$

and, for every $k \in \mathbb{N}$ and $n \in \{1, \ldots, N\}$, the weights in algorithm (5.5) are given by $\lambda_{n,k} = \lambda ((x^{(n)})^2 + \varepsilon)^{-1}$.

5.3 Nonconvex norms

In this section we show that our method can be used to handle nonconvex $\ell_\rho^p$-norms, where $\rho \in [0, 1]$, defined by

$$(\forall x \in \mathbb{R}^N) \quad \ell_\rho^p(x) = \sum_{n=1}^{N} |x|^p. \quad (5.11)$$

The proximity operator (relative to the Euclidean norm) of $\ell_\rho^p$, relative to the metric induced by the diagonal matrix $A = \text{Diag}(\{a^{(n)}\}_{1 \leq n \leq N})$, has an explicit formula, however it necessitates to find roots of polynomial equations [8]:

$$(\forall x \in \mathbb{R}^N) \quad \text{prox}_{\ell_\rho^p}^A(x) = (\mathcal{P}^{(n)})_{1 \leq n \leq N}$$

where $$(\forall n \in \{1, \ldots, N\}) \quad \mathcal{P}^{(n)} = \begin{cases} 0, & \text{if } |x^{(n)}|^{p-2} > \frac{a^{(n)}}{2-\rho} \left( \frac{2-\rho}{2} \right)^{1-\rho} \\ \frac{t^{(n)} x^{(n)}}{t^{(n)} x^{(n)}}, & \text{otherwise}. \end{cases} (5.12)$$
where \( t^{(n)} > 0 \) is such that \( (a^{(n)})^{-1}|x|^p - 2(\rho - 1)(t^{(n)})^{\rho - 1} + t - 1 = 0 \). In practice \((t^{(n)})_{1 \leq n \leq N}\) is found approximately using the Newton method [14].

The proposed approach cannot handle directly the \( \ell_\rho^p \) function, and we need to introduce an approximation of it. Let, for every \( p \in \{1, \ldots, N\} \), \( \psi_p \) be the function defined in (5.6). In this case, to obtain the \( \ell_\rho^p \) function defined in (5.11), we should choose, for every \( p \in \{1, \ldots, P\} \) and \( u \in [0, +\infty[ \), \( \phi_p(u) = u^\rho \). However this function is not differentiable at 0. Thus, instead we propose to choose

\[
(\forall u \in [0, +\infty[) \quad \phi_p(u) = (u + \varepsilon)^\rho - \varepsilon^\rho,
\]

where \( \varepsilon > 0 \). Then, we have

\[
(\forall x \in \mathbb{R}^N) \quad g(x) = \lambda \tilde{\ell}_{p,\varepsilon}(Wx)^\rho,
\]

where \( \tilde{\ell}_{p,\varepsilon} \) is a slightly modified version of the nonconvex \( \ell_\rho^p \)-norm defined in (5.11):

\[
(\forall u \in \mathbb{R}^P) \quad \tilde{\ell}_{p,\varepsilon}(u)^\rho = \sum_{p=1}^{P} (|u(p)| + \varepsilon)^\rho - \varepsilon^\rho.
\]

In this context, for every \( p \in \{1, \ldots, P\} \) and \( k \in \mathbb{N} \), we have

\[
(\forall x \in \mathbb{R}^N) \quad q_p(x, x_k) = \lambda_p \left( (|Wx_k(p)| + \varepsilon)^\rho - \varepsilon^\rho \right) + \lambda_{p,k} \left( |Wx|^\rho - |Wx_k(p)| \right),
\]

where \( \lambda_{p,k} = \lambda_p \rho (|Wx_k(p)| + \varepsilon)^{\rho - 1} \). Therefore, the proposed algorithm given in (3.2) reduces to

\[
x_0 \in \text{dom } g,
\]

for \( k = 0, 1, \ldots \)

\[
\Lambda_k = \text{Diag} \left( \lambda_p \rho (|Wx_k(p)| + \varepsilon)^{\rho - 1} \right)_{1 \leq p \leq P}
\]

\[
\tilde{x}_{k,0} = x_k,
\]

for \( i = 0, \ldots, I_k - 1 \)

\[
\tilde{x}_{k,i+1} = \text{prox}_{\gamma_{k,i} A_{k,i}} \left( \tilde{x}_{k,i} - \gamma_{k,i} A_{k,i}^\top \nabla h(\tilde{x}_{k,i}) \right),
\]

\[
x_{k+1} = \tilde{x}_{k,I_k}.
\]

In Section 6.2 we provide simulation results showing the good behavior of the proposed approach compared with using a classic VMFB algorithm where the proximity operator is computed using (5.12).

### 6 Simulations

Many imaging problems such as reconstruction, restoration, inpainting, etc., can be formulated as inverse problems. In this context, the objective is to find an estimate \( x^* \in \mathbb{R}^N \) of an original unknown image \( x \in \mathbb{R}^N \) from degraded observations \( y \in \mathbb{R}^M \), given by \( y = Hx + b \), where
$H: \mathbb{R}^N \to \mathbb{R}^M$ is a linear observation operator, and $b \in \mathbb{R}^M$ is a realization of an additive independent identically distributed (i.i.d.) random noise. When the random noise is normally distributed, with zero-mean, a common approach to find $x^\ast$ is to define it as the minimizer of a penalized least-squares criterion, i.e. solve (1.1) with, for every $x \in \mathbb{R}^N$, $h(x) = \frac{1}{2}\|Hx - y\|^2$.

In the remaining of the section, we will consider a toy restoration example in image processing. Precisely, we choose $x$ to be the image jetplane of size $N = 256 \times 256$ shown in Figure 1, and $H \in [0, +\infty)^{N \times N}$ to model a blurring operator. In this case, $H$ is implemented as a convolution operator such that the image is convolved with a motion blurring kernel of length 5 and angle $60^\circ$. Note that diagonal preconditioning matrices can be computed in this case using, e.g. [15]. Nevertheless, since the preconditioning scheme has been already widely discussed in the literature (see e.g. [15, 16, 19, 42]), we will not investigate this functionality in our simulations. The noisy observation is then obtained building $b$ as a realization of an i.i.d. Gaussian variable with zero mean and standard deviation $\sigma > 0$. In our simulations we will consider two different noise levels, defined through the input signal-to-noise (iSNR): $\text{iSNR} = \log_{10} \frac{\|Hx\|^2}{(N\sigma^2)}$. Precisely, we will consider the cases when $\text{iSNR} = 20$ dB and $\text{iSNR} = 25$ dB. For each case, we run simulations for 50 realizations of random noise. In Figure 1 are shown the blurred image $Hx \in \mathbb{R}^N$ and an example of a noisy observation $y \in \mathbb{R}^N$, when $\text{iSNR} = 25$ dB.

We will consider two different regularization terms to find an estimate of $x$, the log-sum penalization and the $\ell^p_\rho$ penalization described in Sections 5.2.1 and 5.3, respectively. For the two considered penalization terms we will provide reconstruction results obtained using a basic VMFB algorithm [15] and the proposed C2FB algorithm. The objective is two-fold. Firstly, we will compare the reconstruction quality of the estimate provided by each of the methods. To evaluate the reconstruction quality of the estimate $x^\ast$, we use the signal-to-noise (SNR) ratio, which is defined, for an image $x \in \mathbb{R}^N$ as $\text{SNR} = 10\log_{10} \frac{\|x\|^2}{\|x - x^\ast\|^2}$. Secondly, we will investigate the convergence speed of both the methods, in terms of number of iterations needed to reach convergence. We consider that both the algorithms have converged when the following stopping criteria are fulfilled:

$$
\begin{align*}
\|x_k - x_{k+1}\| &< 10^{-6}\|x_{k+1}\|, \\
\|f(x_k) - f(x_{k+1})\| &< 10^{-5}\|f(x_{k+1})\|,
\end{align*}
$$

(6.1)

where $(x_k)_{k \in \mathbb{N}}$ is the sequence generated either by the standard VMFB algorithm or by the proposed approach. For both the evaluation criteria we will show that the proposed approach leads to better results than the VMFB algorithm. Finally, since the proposed approach is proved to converge for any value of $I_k \in \mathbb{N}^*$, we consider different values of $I_k \equiv I \in \{1, \ldots, 250\}$. This will allow us to show that there is an optimal value for $I_k$ in terms of convergence speed, suggesting that there is no need for reaching convergence in the inner-iterations before recomputing the majorant function.
6.1 Log-sum penalization

In this section we present the simulation results obtained when solving the problem described above, using the log-sum regularization described in Section 5.2.1. More precisely, we propose to minimize

$$\min_{x \in \mathbb{R}^N} \left\{ f(x) := \frac{1}{2} \|Hx - y\|^2 + \lambda \sum_{n=1}^{N} \log(|[W_x]|(n)) + \varepsilon) \right\},$$  

(6.2)

where $\lambda > 0$, $\varepsilon > 0$, and $W: \mathbb{R}^N \to \mathbb{R}^N$ models the Db8 wavelet transform [21], with 4 decomposition levels. As explained in Section 5.2.1, this problem can be solved using the VMFB algorithm, where the proximity operator of the log-sum function is given by (5.9). Alternatively, it can be solved using our method, as described in algorithm (5.8). In this section, we provide the results obtained using both the methods, comparing the quality reconstruction (SNR), and the number of iterations needed to reach convergence. Since the minimization problem is non-convex, the two algorithms may not converge to the same solution. In particular, we have observed that the regularization parameter $\lambda > 0$ giving the best reconstruction quality is not the same for the two different methods. Precisely, in the proposed reweighting approach, we have chosen $\lambda = 120$ and $\lambda = 60$ for the cases with iSNR = 20 dB and iSNR = 25 dB, respectively. Similarly, for the standard VMFB algorithm with exact proximity operator, we have chosen $\lambda = 3000$ and $\lambda = 800$ for the cases with iSNR = 20 dB and iSNR = 25 dB, respectively.

Results are given in Figure 2, considering (top) iSNR = 20 dB and (bottom) iSNR = 25 dB. In these cases, the SNR of the observed images $y$ are equal approximately to 18 dB and 21 dB, respectively. The blue curves are obtained by the proposed approach. The two plots in each figure represent (left) the SNR value of the estimate $x^*$ as a function of $I$; and (right) the total number of iterations needed to reach convergence as a function of $I$. The total number of iterations is given by $K^* \times I$, where $K^*$ is the number of iterations computed in the outer-loop in algorithm (5.8) to satisfy the stopping criteria given in (6.1). The red curves are obtained considering the standard VMFB algorithm, computing exactly the proximity operator. In this case all the curves are constant as there is no inner-loop in the algorithm. For both methods, the continuous lines represent the mean values, and the dotted lines show the associated results within 1 standard deviation around the mean.

Figure 1: Image jetplane used for the simulations. From left to right: Original unknown image $x$, blurred image $Hx$, and noisy observation $y$ for an iSNR of 25 dB. The SNR between the original image $x$ and the observations $y$ is equal to 20.9 dB.
Figure 2: Results for the log-sum penalization considering a noise level with (top) iSNR = 20 dB (i.e. SNR of $y$ equal $\approx 18$ dB) and (bottom) iSNR = 25 dB (i.e. SNR of $y$ equal $\approx 21$ dB). Comparison between the results obtained with the proposed reweighting $\ell_1$ algorithm (5.8) considering different numbers of iterations in the inner-loop $I_k \equiv I \in \{1, \ldots, 250\}$ (blue curves), and the results obtained using a VMFB algorithm with the exact proximity operator of the log-sum penalization computed as per (5.9) (red curves). From left to right: SNR values in dB, and global number of iterations needed to reach convergence. For the proposed method, the global number of iterations needed to reach convergence corresponds to $K^* \times I_k$, where $K^* \in \mathbb{N}$ is the number of iterations for the outer-loop in algorithm (5.8). Note that the red curves are constant since when the proximity operator is computed exactly, there is no inner-iterations for the reweighting. The continuous lines are the mean values obtained over 50 realization of random noise, and the dotted lines show the associated results within 1 standard deviation around the mean.

For both the considered noise levels, we can observe that the reweighting approach leads to better reconstruction results in terms of SNR when $I_k = I > 2$ than the standard VMFB with exact proximity operator. Precisely, when iSNR = 20 dB (resp. iSNR = 25 dB), the proposed approach leads to results of SNR $\approx 22.1$ dB (resp. SNR $\approx 23.6$ dB), while the standard VMFB algorithm leads to results with SNR $= 21.6$ dB (resp. SNR $= 23.2$ dB). In addition, in average, our method necessitates less global iterations to reach convergence when $I \geq 5$. For the proposed approach, we can observe that there is an optimal value for $I$ in terms of total iteration number, while the SNR value is almost constant. Precisely, when iSNR = 20 dB (resp. iSNR = 25 dB), the optimal value is around $I = 10$ (resp. $I = 20$), for an averaged total number of iterations of $K^* \times I = 124$ (resp. $K^* \times I = 301$). In comparison, the standard VMFB algorithm necessitates 270 (resp. 473) iterations to converge. This observation suggests that reducing the number of iterations in the inner-loop (instead of reaching convergence in each inner-loop before re-computing the weights, as suggested in classical reweighting $\ell_1$ algorithms [11]) can accelerate the convergence of the reweighting $\ell_1$ algorithm without altering the reconstruction quality.

26
6.2 $\ell_\rho$ penalization

In this section we present the simulation results obtained when solving the problem described at the beginning of the section, using the $\ell_\rho$ regularization described in Section 5.3, with $\rho = 0.1$. More precisely, we propose to

$$\minimize_{x \in \mathbb{R}^N} \left\{ f(x) := \frac{1}{2} \|Hx - y\|^2 + \lambda \tilde{\ell}_{0.1,\varepsilon}(Wx)^{0.1} \right\}, \quad (6.3)$$

where $\lambda > 0$, $\varepsilon > 0$, $\tilde{\ell}_{0.1,\varepsilon}$ is the approximation of the $\ell_{0.1}$ norm defined in (5.15), and $W : \mathbb{R}^N \to \mathbb{R}^N$ models the Db8 wavelet transform [21], with 4 decomposition levels. As explained in Section 5.3, this problem can be solved using our method, with algorithm (5.17). In the limit case when $\varepsilon = 0$, then $\tilde{\ell}_{0.1,\varepsilon}$ reduces to the exact $\ell_{0.1}$ norm, and problem (6.3) can be solved using the VMFB algorithm, where the proximity operator of the $\ell_{0.1}$ norm is given by (5.12). Note that even if the proximity operator of the $\ell_{0.1}$ norm has an explicit formula, in practice it requires to compute sub-iterations. As for the log-sum regularization, in this section, we provide the results obtained using both methods, comparing the quality reconstruction (SNR), and the number of iterations needed to reach convergence. Concerning the regularization parameter $\lambda$, it has been chosen for each method in order to obtain the best quality results. Precisely, in the proposed approximated approach, we have chosen $\lambda = 60$ and $\lambda = 30$ for the cases with iSNR $= 20$ dB and iSNR $= 25$ dB, respectively. Similarly, for the standard VMFB algorithm with exact proximity operator, we have chosen $\lambda = 5000$ and $\lambda = 1200$ for the cases with iSNR $= 20$ dB and iSNR $= 25$ dB, respectively.

Results are given in Figure 3, considering (top) iSNR $= 20$ dB and (bottom) iSNR $= 25$ dB (the SNR of the observed images $y$ are equal approximately to 18 dB and 21 dB, respectively). The blue curves are obtained using the proposed approach. The two plots in each figure represent (left) the SNR value of the estimate $x^\star$ as a function of $I$; and (right) the total number of iterations needed to reach convergence as a function of $I$. The total number of iterations is obtained by $K^\star \times I$, where $K^\star$ is the number of iterations computed in the outer-loop in algorithm (5.17) to satisfy the stopping criteria given in (6.1). The red curves are obtained considering the standard VMFB algorithm, computing the proximity operator using equation (5.12). In this case all the curves are constant as there is no inner-loop in the algorithm. For both methods, the continuous lines represent the mean values, and the dotted lines show the associated results within 1 standard deviation around the mean.

Similarly to the observations we made for the log-sum regularization, for both the considered noise levels, the proposed approach leads to better SNR values when $I_k = I > 2$ than the standard VMFB algorithm with the exact $\ell_{0.1}$-norm. More precisely, when iSNR $= 20$ dB (resp. iSNR $= 25$ dB), the proposed approach leads to results of SNR $\approx 22.2$ dB (resp. SNR $\approx 23.6$ dB), while the standard VMFB algorithm leads to results with SNR $= 21.4$ dB (resp. SNR $= 23.3$ dB). The results for the number of iterations is slightly different from the one obtained considering the log-sum regularization. On the one hand, when iSNR $= 20$ dB, the optimal value for $I$ is around
Figure 3: Results for the $\ell_\rho$ penalization, with $\rho = 0.1$, considering a noise level with (top) iSNR = 20 dB (i.e. SNR of $y$ equal \approx 18 dB) and (bottom) iSNR = 25 dB (i.e. SNR of $y$ equal \approx 21 dB). Comparison between the results obtained with the proposed approximation algorithm (5.17) considering different numbers of inner-iterations $I_k \equiv I \in \{1, \ldots, 250\}$ (blue curves), and the results obtained using a VMFB algorithm with the exact proximity operator of the $\ell_\rho$ penalization computed as per (5.12) (red curves). From left to right: SNR values in dB, and global number of iterations needed to reach convergence. For the proposed method, the global number of iterations corresponds to $K^* \times I$, where $K^* \in \mathbb{N}$ is the number of outer-iterations in algorithm (5.8). The red curves are constant since when the proximity operator is computed exactly, there is no inner-iterations for the reweighting. The continuous lines are the mean values obtained over 50 realization of random noise, and the dotted lines show the associated results within 1 standard deviation around the mean.

For a total number of iterations equal to $K^* \times I = 124$. In comparison, the standard VMFB algorithm requires in average 144 iterations to reach convergence. In this case, the acceleration of the proposed approach is less obvious than for the log-sum regularization. In addition, the proposed approach requires less iterations than the standard VMFB algorithm only when $I \in \{10, \ldots, 30\}$. It is worth noticing however that these results are balanced by the fact that our method does not require sub-iterations for the computation of the proximity operator, unlike the standard VMFB algorithm. On the other hand, when iSNR = 25 dB, the proposed approach necessitates less iterations to converge when $I \in \{3, \ldots, 230\}$. The optimal choice is $I = 30$, which leads to an average number of iterations for convergence equal to $K^* \times I = 282$. In comparison, the standard VMFB algorithm requires an average of 451 iterations. Therefore, we can conclude that the proposed approach provides a good alternative to the classic VMFB method, in terms of both quality reconstruction and convergence speed.

References

[1] H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. Math. Program., 116:5–16, 2008.
[2] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program.*, 137:91–129, Feb. 2011.

[3] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[4] J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM J. Optim.*, 17:1205–1223, 2006.

[5] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM J. Optim.*, 18(2):556–572, 2007.

[6] J. Bolte, A. Daniilidis, O. Ley, and L. Mazet. Characterizations of Łojasiewicz inequalities: subgradient flows, talweg, convexity. *Trans. Amer. Math. Soc.*, 362(6):3319–3363, 2010.

[7] J. Bolte and E. Pauwels. Majorization-minimization procedures and convergence of sqp methods for semi-algebraic and tame programs. *Math. Oper. Res.*, 41:442–465, 2016.

[8] Kristian Bredies, Dirk, and A. Lorenz. Iterated hard shrinkage for minimization problems with sparsity constraints. *SIAM Journal on Scientific Computing*, 30(2):657–683, 2008.

[9] M. Burger, A. Sawatzky, and G. Steidl. *First Order Algorithms in Variational Image Processing*, pages 345–407. Springer International Publishing, 2016.

[10] J. V. Burke. Descent methods for composite nondifferentiable optimization problems. *Math. Program.*, 33(3):260–279, 1985.

[11] Emmanuel J Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, volume 3, pages 1433–1452. Madrid, Spain, 2006.

[12] C. Cartis, N. I. M. Gould, and P. L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. *SIAM J. Optim.*, 21(4):1721–1739, 2011.

[13] G. H.-G. Chen and R. T. Rockafellar. Convergence rates in forward-backward splitting. *SIAM J. Optim.*, 7(2):421–444, 1997.

[14] G. Chierchia, E. Chouzenoux, P. L. Combettes, and J.-C. Pesquet. The proximity operator repository. user’s guide. Technical report. http://proximity-operator.net.

[15] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *J. Optim. Theory Appl.*, 162(1), Jul. 2014.

[16] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. A block coordinate variable metric forward-backward algorithm. *J. Global Optim.*, 66(3):457–485, Nov. 2016.
[17] P. L. Combettes. *The Convex Feasibility Problem in Image Recovery*, volume 95 of *Advances in Imaging and Electron Physics*. Academic Press, New York, 1996.

[18] P. L. Combettes, D. Dung, and B. C. Vu. Proximity for sums of composite functions. *J. Math. Anal. Appl.*, 380(2):680–688, Aug. 2011.

[19] P. L. Combettes and B. C. Vu. Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):1289–1318, Sep. 2014.

[20] Patrick L. Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[21] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4:247–269, 1998.

[22] D. Drusvyatskiy, and A. S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. Technical report, 2016. arXiv:1602.06661.

[23] D. Drusvyatskiy, A.d. Ioffe, and A. S. Lewis. Nonsmooth optimization using taylor-like models: error bounds, convergence, and termination criteria. Technical report, 2016. arXiv:1610.03446.

[24] R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. In *Nondifferential and Variational Techniques in Optimization*, pages 67–76. Springer, 2009.

[25] P. Frankel, G. Garrigos, and J. Peypouquet. Splitting methods with variablemetric for kurdyka-ojasiewicz functions and general convergence rates. *J. Optim. Theory Appl.*, 165(3):874–900, 2015.

[26] J. Geipring and M. Moeller. Composite optimization by nonconvex majorization-minimization. *SIAM J. Imaging Sci.*, 11(4):2494–2598, 2018.

[27] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, New York, 1993.

[28] D. R. Hunter and K. Lange. A tutorial on mm algorithms. *Amer. Statist.*, 58:30–37, 2004.

[29] K. Kurdyka and A. Parusinski. $w_f$-stratification of subanalytic functions and the Łojasiewicz inequality. *Comptes rendus de l’Académie des sciences. Série 1, Mathématique*, 318(2):129–133, 1994.

[30] A. S. Lewis and S. J. Wright. A proximal method for composite minimization. *Math. Program.*, 158:501–546, 2015.

[31] J. Liang and C.-B. Schönlieb. Improving “fast iterative shrinkage-thresholding algorithm”: Faster, smarter and greedier. Technical report, 2019. arXiv:1811.01430.

[32] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16:964–979, 1079.
[33] S. Łojasiewicz. Une propriété topologique des sous-ensembles analytiques réels, pages 87–89. Editions du centre National de la Recherche Scientifique, 1963.

[34] J. Mairal. Optimization with first-order surrogate functions. In Proceedings of the 30th International Conference on Machine Learning - Volume 28, ICML’13, pages III–783–III–791, Atlanta, GA, 2013.

[35] S. Mallat. A Wavelet Tour of Signal Processing. Academic Press, Burlington, MA, 2nd edition, 2009.

[36] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. Bulletin de la Société mathématique de France, 93:273–299, 1965.

[37] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock. On iteratively reweighted algorithms for non-smooth nonconvex optimization in computer vision. SIAM J. Imaging Sci., 8(1):331–372, 2015.

[38] P Ochs, J. Fadili, and T. Brox. Non-smooth non-convex bregman minimization: Unification and new algorithms. J. Optim. Theory. Appl., 181(1):244–278, 2019.

[39] M. J. D. Powell. General algorithms for discrete nonlinear approximation calculations. In Approximation theory, IV, pages 187–218. Academic Press, New York, 1983.

[40] M. J. D. Powell. On the global convergence of trust region algorithms for unconstrained minimization. Math. Program., 29(3):297–303, 1984.

[41] A. Repetti, E. Chouzenoux, and J.-C. Pesquet. A preconditioned forward-backward approach with application to large-scale nonconvex spectral unmixing problems. In Proceedings of the 39th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014), pages 1498–1502, Florence, Italy, 4-9 May 2014.

[42] A. Repetti, M. Q. Pham, L. Duval, E. Chouzenoux, and J.-C. Pesquet. Euclid in a Taxicab: Sparse blind deconvolution with smoothed $\ell_1/\ell_2$ regularization. IEEE Signal Process. Lett., 22(5):539–543, May 2015.

[43] R. T. Rockafellar. Convex Analysis. Princeton University Press, 1970.

[44] R. T. Rockafellar and R. J.-B. Wets. Variational Analysis, volume 317 of Grundlehren der Mathematischen Wissenschaften. Springer, Berlin, 3rd edition, 2009.

[45] Y. Sun, P. Babu, and D. P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. IEEE Trans. Signal Process., 65:794–816, 2017.

[46] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. SIAM J. Control Optim., 38:431–446, 2000.

[47] S. J. Wright. Convergence of an inexact algorithm for composite nonsmooth optimization. IMA J. Numer. Anal., 10(3):299–321, 1990.
[48] C. F. J. Wu. On the convergence properties of the em algorithm. *Annals Statist.*, 11:95–103, 1983.

[49] Y. Yuan. On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Math. Program.*, 31(3):269–285, 1985.