An Extension of Averaged-Operator-Based Algorithms

Miguel Simões
Instituto de Telecomunicações
Instituto Superior Técnico, Univ. Lisboa
Lisbon, Portugal
miguel.simoes@lx.it.pt

José Bioucas-Dias
Instituto de Telecomunicações
Instituto Superior Técnico, Univ. Lisboa
Lisbon, Portugal
bioucas@lx.it.pt

Luis B. Almeida
Instituto de Telecomunicações
Instituto Superior Técnico, Univ. Lisboa
Lisbon, Portugal
luis.almeida@lx.it.pt

Abstract—Many of the algorithms used to solve minimization problems with sparsity-inducing regularizers are generic in the sense that they do not take into account the sparsity of the solution in any particular way. However, algorithms known as semismooth Newton are able to take advantage of this sparsity to accelerate convergence. We show how to extend these algorithms in different directions, and study the convergence of the resulting algorithms by showing that they are a particular case of an extension of the well-known Krasnosel’ski˘ı–Mann scheme.

Index Terms—Convex nonsmooth optimization, primal–dual optimization, semismooth Newton method, forward–backward method, variable metric

I. INTRODUCTION

A. Background

The objective functions of many signal-processing problems can be formulated as sums of two proper lower-semicontinuous convex functions: one that is smooth, \( f : \mathbb{R}^n \to (-\infty, +\infty) \), and another one that need not be smooth, \( g : \mathbb{R}^n \to (-\infty, +\infty) \). The resulting problem is

\[
\text{minimize } f(x) + g(x).
\]

Such problems are typically large-scale and can be solved by using splitting methods, which convert (1) into a sequence of separable subproblems. The (relaxed) forward–backward method [1], [2] is an example of such methods. Its iterations can be broken into a gradient (forward) step on \( f \) and a proximal (backward) step on \( g \), performed consecutively—see Algorithm 1, where \( \text{prox}_{\tau g} \) denotes the proximal operator of function \( g \), i.e.,

\[
\text{prox}_{\tau g}(x) \triangleq \arg\min_{u \in \mathbb{R}^n} \{ g(u) + \frac{1}{2\tau} \| x - u \|_2^2 \}.
\]

When analyzing the properties of many of these and other algorithms, it can be advantageous to use the theory of monotone operators [4]. Let \( 2^\mathbb{R}^n \) denote the power set of \( \mathbb{R}^n \). A set-valued operator \( A : \mathbb{R}^n \to 2^\mathbb{R}^n \) is said to be monotone if \( \langle u - v, x - y \rangle \geq 0 \) for all \( (x, u) \in \text{gra } A \) and \( (y, v) \in \text{gra } A \), where \( \text{gra } A \) denotes the graph of \( A \), and it is said to be maximally monotone if there exists no other monotone operator whose graph properly contains \( \text{gra } A \). Monotone operators are connected to optimization problems as follows. Take, for example, (1). According to Fermat’s rule, its solutions should satisfy the inclusion \( 0 \in \nabla f(x) + \partial g(x) \), where the set-valued operator \( \partial g : \mathbb{R}^n \to 2^{2^\mathbb{R}^n} : x \rightarrow \partial g(x) \) denotes the subdifferential of \( g \) (in the sense of Moreau and Rockafellar [5, Chapter 23]). The operators \( \nabla f \) and \( \partial g \) are examples of maximally-monotone operators [6, Theorem 20.40]. Problem (1) can be seen as a particular case of the problem of finding a zero of the sum of two monotone operators \( A \) and \( C \), i.e.,

\[
\text{find } x \in \mathbb{R}^n \text{ such that } 0 \in A(x) + C(x),
\]

if one makes \( A = \partial g \) and \( C = \nabla f \). Problem (2) may be solved using a generalized version of Algorithm 1, in which Line 5 is replaced with

\[
x^{k+1} \leftarrow x^k + \lambda^k \left( \text{prox}_{\tau A} \left( x^k - \tau C \left( x^k \right) \right) - x^k \right),
\]

where \( J_{\tau A} \triangleq (\text{Id} + \tau A)^{-1} \) is the resolvent of operator \( A \) and \( \text{Id} \) denotes the identity operator. Note that \( J_{\tau \partial g} = \text{prox}_{\tau g} \) [6, Example 23.3].

Algorithm 1: Relaxed forward–backward method.

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Choose ( x^0 \in \mathbb{R}^n ), ( \tau &gt; 0 );</td>
</tr>
<tr>
<td>2</td>
<td>( k \leftarrow 1 );</td>
</tr>
<tr>
<td>3-while</td>
<td>stopping criterion is not satisfied do</td>
</tr>
<tr>
<td>4</td>
<td>Choose ( \lambda^k &gt; 0 );</td>
</tr>
<tr>
<td>5</td>
<td>( x^{k+1} \leftarrow x^k + \lambda^k \left( \text{prox}_{\tau g} \left( x^k - \tau \nabla f \left( x^k \right) \right) - x^k \right) );</td>
</tr>
<tr>
<td>6</td>
<td>( k \leftarrow k + 1 );</td>
</tr>
<tr>
<td>7 end</td>
<td></td>
</tr>
</tbody>
</table>

Problem (2) can alternatively be written as the problem of finding a fixed point of the operator \( R \triangleq J_{\tau A} \circ (\text{Id} - \tau C) \):

\[
\text{find } x \in \mathbb{R}^n \text{ such that } R(x) = x.
\]

In general, the solutions of a convex optimization problem correspond to the fixed points of a certain operator, and an iterative optimization algorithm corresponds to a fixed-point method. We can rewrite (3) as

\[
x^{k+1} \leftarrow T_{\lambda^k} \left( x^k \right) \triangleq x^k + \lambda^k (R(x^k) - x^k),
\]

We say that an operator \( R : \mathbb{R}^n \to \mathbb{R}^n \) is nonexpansive if \( \| u - v \| \leq \| x - y \| \) for all \( (x, u) \in \text{gra } R \) and \( (y, v) \in \text{gra } R \). Let \( R \) be a generic nonexpansive operator and let \( \lambda \in [0, 1] \). Then

This work was supported by the Fundação para a Ciência e Tecnologia within the Portuguese Ministry for Science, Technology and Higher Education under Project UID/EEA/50008/2013 and Grant BPD/N.º 134 - 16/10/2017.
the operator $T \triangleq (\mathrm{Id} - \lambda) + \lambda R$ is said to be $\lambda$-averaged. It obeys the following contractive property [6, Proposition 4.25]:

$$\|T(x) - T(y)\|^2 \leq \|x - y\|^2 - \frac{1 - \lambda}{\lambda} \|(\mathrm{Id} - T)(x) - (\mathrm{Id} - T)(y)\|^2$$  \hspace{1cm} (6)

for all $x, y \in \mathbb{R}^n$. In particular, when $\lambda = 1/2$, $T$ is said to be firmly nonexpansive. The resolvents of maximally-monotone operators are firmly-nonexpansive [6, Corollary 23.8]. Iteration (5) is known as the Krasnosel’skiĭ–Mann scheme and is the basis of not only the forward–backward method but also other optimization algorithms, such as the Douglas–Rachford one [4], [6]. It can be shown that, under certain conditions, the Krasnosel’skiĭ–Mann scheme converges to $\text{Fix } R$, where $\text{Fix } R$ denotes the set of fixed points of $R$.

The convergence rate of the forward–backward method (Algorithm 1) can be shown to be sublinear, or, under certain assumptions, to be linear. This rate can often be improved by incorporating second-order information about $f$ if this function is twice-differentiable. The local convergence rate of second-order methods is superlinear or even quadratic. As an example, consider the second-order version of Algorithm 1, which is given by replacing Line 5 with the iteration $x^{k+1} \leftarrow x^k + \lambda^k \left(\text{prox}^B_{g^k} \left( x^k - [B^k]^{-1} \nabla f (x^k) \right) - x^k \right)$ [7]-[9], where $B^k$ is a positive-definite (PD) matrix (the Hessian of $f$ or an approximation of it) and $\text{prox}^B_{g^k}$ denotes the proximal operator of $g$ relative to the norm $\| \cdot \|^2_{B^k}$, i.e.,

$$\text{prox}^B_{g^k}(x) \triangleq \arg\min_{u \in \mathbb{R}^n} \{g(u) + \frac{1}{2}\|u - x\|^2_{B^k}\}.$$  \hspace{1cm} (7)

More generally, and from an operator-centric perspective, by using second-order methods such as these, one is actually solving a left-preconditioned version of (2), in the sense that instead of directly tackling that problem we are considering problems that share the same set of solutions but may be more convenient to solve:

$$\text{find } x \in \mathbb{R}^n \text{ such that } 0 \in UA(x) + UC(x),$$  \hspace{1cm} (8)

where $U$ is a PD operator. In what follows, we denote positive definiteness by $U \succ 0$ and positive semidefiniteness by $U \succeq 0$.

### C. Notation and outline

A detailed account of the notions listed in this section can be found in the work of Bauschke and Combettes [6]. We denote the scalar product of a Hilbert space by $\langle \cdot, \cdot \rangle$ and the associated norm by $\| \cdot \|$. The range of an operator $A$ is denoted by $\text{ran } A$, and the adjoint of $A$ by $A^*$. We say that an operator $A : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with constant $L > 0$ if $\|u - v\| \leq L\|x - y\|$, for all $(x, u) \in \text{gra } A$ and $(y, v) \in \text{gra } A$. Additionally, let $\Gamma_0(\mathbb{R}^n)$ denote the class of all proper lower-semicontinuous convex functions from $\mathbb{R}^n$ to $]-\infty, +\infty]$.

### II. SEMISMOOTH NEWTON METHODS

Semismooth Newton methods were originally developed with the goal of using Newton-like methods to minimize certain nonsmooth functions at a superlinear convergence rate. To illustrate why these methods may be useful when solving problems of the form of (1), consider, as an example, that $f = \|y - \mathbf{H} \cdot x\|^2$, and $g = \mu \| \cdot \|_1$, where $y \in \mathbb{R}^m$, $\mathbf{H} \in \mathbb{R}^{m \times n}$, and $\mu > 0$. For problems such as these, it was shown by Hintermüller [12] that some semismooth Newton methods are equivalent to some active-set methods. As we discuss in Section IV, the fact that these methods can be written as active-set ones allows for significant time savings when solving certain problems, namely the ones involving sparsity-inducing regularizers, as is the case of the $\ell_1$ norm.

Let $G : \mathbb{R}^n \to \mathbb{R}^n$ be an operator such that $G : x \to x - \text{prox}_{\mu \| \cdot \|_1}(x - 2\mu \mathbf{H}^* (\mathbf{H}x - y))$. The solution of the problem under consideration should satisfy the nonlinear equation $G(x) = 0$, which is nonsmooth, since $\text{prox}_{\mu \| \cdot \|_1}$ is not everywhere differentiable. There are, however, generalizations of the concept of differentiability that are applicable...
to an operator such as $G$. One of them is the B(ouligand)-
differential [13, Definition 4.6.2], which is defined as follows.
Suppose that a generic operator $G : D \subset \mathbb{R}^n \to \mathbb{R}^m$ is locally
Lipschitz, where $D$ is an open subset. Then by Rademacher’s
corollary, $G$ is differentiable almost everywhere in $D$. Let $C$
denote the subset of $\mathbb{R}^n$ consisting of the points where $G$
is differentiable (in the sense of Fréchet [6, Definition 2.45]). The
B-differential of $G$ at $x$ is $\partial_B G(x) \triangleq \left\{ \lim_{x_k \to x } \nabla G(x_k) \right\}$, where $\{x_k\}$ is a sequence such that $x_k \in C$ for all $j$ and
$\nabla G(x_k)$ denotes the Jacobian of $G$ at $x_k$.

The B-differential of an operator at a given point may not
be unique: for example, take $\text{prox}_{\mu\|\cdot\|}(x)$, which can be
evaluated element-wise by computing $\max \{ \|x\|/\mu \} \circ
\text{sgn}(\{x\})$ for $i \in \{1, \ldots, n\}$. A possible $H \in \partial_B \text{prox}_{\mu\|\cdot\|}(x)$
is a binary diagonal matrix defined as [14, Proposition 3.3]
$$[H]_{ii} = \begin{cases} 1 & \text{if } \|x_i\| > \mu, \\
0 & \text{otherwise}. \end{cases} \quad (10)$$

This generalization of the concept of differentiability can also be used to formulate the so-called semismooth Newton
method, which is characterized by the iteration $x^{k+1} \leftarrow
x^k - [H]^{-1} G(x^k)$, where $H \in \partial_B G(x^k)$. It can be shown that
this method locally converges superlinearly for operators known as semismooth [15]. Let $x \in D$ and $d \in \mathbb{R}^n$;
semismooth operators are operators that are directionally dif-
ferentiable at a neighborhood of $x$ and that, for any $H \in
\partial_B G(x + d)$, satisfy the condition $H d - G'(x; d) = o(||d||)$
for $d \to 0$, where $G'(x; d)$ denotes the directional derivative
of $G$ at $x$ along $d$. Examples of semismooth functions are
the Euclidean norm and piecewise-differentiable functions [16,
Chapter 2], $\text{prox}_{\mu\|\cdot\|}(x)$ being an example of the latter. Note
that the semismooth Newton method is a particular case of (8),
although we impose that $\text{Id} \succeq \Lambda^k \succ 0$ in the latter equation,
which is not necessarily true for this method.

III. AN EXTENSION OF AVERAGED-OPERATOR-BASED
ALGORITHMS

In this section, we define operator-weighted averaged op-
erators, and show that they have a contraction property. We also
study the asymptotic behavior of fixed-point iterations of these
operators. Such iterations can be seen as an extension of the
Krasnosel’skiǐ–Mann scheme [cf. (5)]. We base our analysis
on the fact that these iterations produce a sequence that is
variable-metric Fejér monotone [17], [18]. We then present
an algorithm that uses operator-weighted averaged operators,
and that solves a primal–dual problem that encapsulates many
problem formulations [10], [18].

A. An extension of the Krasnosel’skiǐ–Mann scheme

Definition III.1 (Operator-weighted averaged operators). Let
$D$ be a nonempty subset of $\mathbb{R}^n$, let $\epsilon \in [0,1]$, and let $\Lambda$ be an
operator in $\mathbb{R}^n$ such that
$$\mu \text{Id} \succeq \Lambda \succeq \alpha \text{Id}, \quad \text{where } \mu, \alpha \in [\epsilon, 1 - \epsilon]. \quad (11)$$
We say that an operator $T_\Lambda : D \to \mathbb{R}^n$ is an operator-weighted averaged operator if there exists a nonexpansive operator $R : D \to \mathbb{R}^n$ such that
$$T_\Lambda \triangleq (\text{Id} - \Lambda) + \Lambda R. \quad (12)$$

We have proved the following results:

Proposition III.2. Let $D$ be a nonempty subset of $\mathbb{R}^n$, let
$\epsilon \in [0,1]$, let $\Lambda$ be an operator in $\mathbb{R}^n$ satisfying (11), let $R : D \to \mathbb{R}^n$ be a nonexpansive operator, and let $T_\Lambda : D \to \mathbb{R}^n$
be an operator as defined in (12). Then the operator $T_\Lambda$
is $\mu$-averaged in the metric induced by $\Lambda^{-1}$. In other words, the
operator $T_\Lambda$ verifies
$$||T_\Lambda(x) - T_\Lambda(y)||^2_{\Lambda^{-1}} \leq ||x - y||^2_{\Lambda^{-1}} - \frac{1 - \mu}{\mu} ||(\text{Id} - T_\Lambda)(x) - (\text{Id} - T_\Lambda)(y)||^2_{\Lambda^{-1}}$$
for all $x, y \in D$.

Theorem III.3. Let $D$ be a nonempty closed convex subset
of $\mathbb{R}^n$, let $\epsilon \in [0,1]$, let $\{\eta^k\} \in L^1_+(\mathbb{N})$, let $\{\Lambda^k\}$ be a sequence of PD operators in $\mathbb{R}^{n \times n}$ such that, for all $k \in \mathbb{N}$,
$$\begin{aligned}
\mu_k \text{Id} &\succeq \Lambda^k \succeq \alpha_k \text{Id}, \\
\mu_k, \alpha_k &\in [\epsilon, 1 - \epsilon],
\end{aligned} \quad (13)$$
and let $R : D \to D$ be a nonexpansive operator such that
Fix $R \neq \emptyset$. Additionally, let $x^k \in D$ and let, for all $k$, $\{x^k\}$
be a sequence generated by (8). Then $\{x^k\}$ converges to a
point in Fix $R$.

B. Primal–dual optimization algorithms

Combettes and Pesquet studied a primal–dual problem that
generalizes many problems [10, Problem 4.1]. By being able
to devise an algorithm to solve this problem, we are effect-
tively tackling a large number of problems simultaneously
(problem (1) is one of these). Let $m, n$, and $N$ be strictly-
positive integers, let $g \in \Gamma_0(\mathbb{R}^n)$, let $\mu \in [0, \infty[$, let
$f : \mathbb{R}^n \to ]-\infty, \infty]$ be convex and differentiable with a
$\mu^{-1}$-Lipschitzian gradient, and let $z \in \mathbb{R}^n$. For every
$j \in \{1, \ldots, N\}$, let $r_j \in \mathbb{R}^m$, let $h_j \in \Gamma_0(\mathbb{R}^m)$,
let $\nu_j \in [0, \infty[$, let $l_j \in \Gamma_0(\mathbb{R}^m)$ be $\nu_j$-strongly convex,\footnote{1A function $l$ is said to be $\nu$-strongly convex if $l - \frac{\nu}{2}(x, x)$ is convex, for
some $\nu > 0$.} let $L_j \in \mathbb{R}^{m \times m}$ such that $L_j \neq 0$, and let $\omega_j$ be real numbers in $[0,1]$ such that $\sum_{j=1}^N \omega_j = 1$. The problem is as follows:

Problem III.4. Solve the primal minimization problem,
$$\begin{aligned}
&\text{minimize } \mu x \in \mathbb{R}^n \quad \sum_{j=1}^N \omega_j (h_j * \text{inf } l_j)(L_j x - r_j) + f(x) - \langle x, z \rangle,
&\text{subject to its corresponding dual minimization problem,}
\end{aligned}$$

$$\begin{aligned}
&\text{minimize } d_1 \in \mathbb{R}^{m_1}, \ldots, d_N \in \mathbb{R}^{m_N} \quad \left( g^* * \text{inf } h^* \right) \left( z - \sum_{j=1}^N \omega_j L_j d_j \right) \\
&\quad + \sum_{j=1}^N \omega_j (h_j^*(d_j) + l_j^*(d_j) + \langle d_j, r_j \rangle).
\end{aligned}$$

The sets of solutions to these primal and dual problems are
denoted by $P$ and $D$, respectively.
Consider Algorithm 2 to solve Problem III.4. In what follows, for all \( j \), \( \{U^k\}, \{\Lambda^k\} \) are sequences of linear operators, and \( \{a^k\}, \{b^k\}, \{c^k\}, \{e^k\} \) are absolutely-summable sequences that can be used to model errors. Algorithm 2 is an extension of [18, Example 6.4].

\textbf{Algorithm 2: An application of (8) to solve Problem III.4.}

1. Choose \( x^0 \in \mathbb{R}^n \) and \( d^0_j \in \mathbb{R}^{m_j} \), \( j = 1, \ldots, N \).
2. \( k \leftarrow 1 \);
3. \textbf{while} stopping criterion is not satisfied \textbf{do}
   4. \hspace{1em} for \( j = 1, \ldots, N \) \textbf{do}
      5. \hspace{2em} Choose \( U_j^k, \Lambda_j^k \succ 0 \) s.t. \( \Lambda_j^k \prec I_d \);
      6. \hspace{2em} \( q_j^k = \text{prox}_{\gamma_j^k}^{-1} (d_j^k + U_j^k (L_j x^k - \nabla L_j^k (d_j^k - e_j^k - r_j^k)) + b_j^k) \);
      7. \hspace{2em} \( y_j^k = 2q_j^k - d_j^k \);
      8. \hspace{2em} \( d_j^{k+1} = d_j^k + \Lambda_j^k (q_j^k - d_j^k) \);
   9. \hspace{1em} \textbf{end}
10. Choose \( U^k, \Lambda^k \succ 0 \) s.t. \( \Lambda^k \prec I_d \);
11. \( p^k = \text{prox}_{\mu\|\cdot\|_1}^{-1} (x^k - U^k (\sum_{j=1}^N \omega_j^k L_j^k y_j^k + \nabla f(x^k) + c^k - z)) + a^k \);
12. \( x^{k+1} = x^k + \Lambda^k (p^k - x^k) \);
13. \( k \leftarrow k + 1 \);
\textbf{end}

following corollary establishes some convergence properties of Algorithm 2.

\textbf{Corollary III.5.} Suppose that
\[ z \in \text{ran} \left( \frac{\partial g + N \sum_{j=1}^N \omega_j L_j^* (\partial h_{j \ast x} \sigma_{j \ast x} \partial h_{j \ast x} (L_j \cdot - r_j) + \nabla f)}{} \right) \]
and set \( \beta \triangleq \min \{\mu, \nu_1, \ldots, \nu_N\} \). Let \( \{U_j^k\} \) be a sequence of PD operators in \( \mathbb{R}^{n \times n} \) and, for every \( j \in \{1, \ldots, N\} \), let \( \{U_j^k\} \) be a sequence of PD operators in \( \mathbb{R}^{m_j \times m_j} \) such that, for all \( k \in \mathbb{N} \),
\[ \mu \tau \text{Id} \succeq U^k \succeq \alpha \text{Id}, \quad \text{Id} \succeq U_j^k \succeq \alpha \text{Id}, \quad \mu, \alpha \in (0, +\infty), \tag{14} \]
let \( \epsilon \in [0, \min\{1, \beta\} \), let \( \{\Lambda^k\} \) be a sequence of PD operators in \( \mathbb{R}^{n \times n} \), and let \( \{\Lambda_j^k\} \) be a sequence of PD operators in \( \mathbb{R}^{m_j \times m_j} \) such that, for all \( k \),
\[ \Lambda^k U^k = U^k \Lambda^k, \quad \Lambda_j^k U_j^k = U_j^k \Lambda_j^k, \tag{15} \]
\[ \mu \text{Id} \succeq \Lambda^k \succeq \alpha \text{Id}, \quad \text{Id} \succeq \Lambda_j^k \succeq \alpha \text{Id}, \quad \mu, \alpha \in (\epsilon, 1], \]
Let, for all \( j \), \( \{a^k\}, \{b^k\}, \{c^k\}, \{e^k\} \in \ell^1(\mathbb{N}) \). For every \( k \), set \( \delta^k \triangleq \left( \sum_{j=1}^N \omega_j \left\| \frac{U_j^k L_j y_j^k}{\sqrt{U_j^k L_j}} \right\|_2^2 \right)^{-1/2} \) and suppose that \( \xi^k \triangleq \frac{\delta^k}{(1 + \delta^k)^{\mu}} \geq \frac{1}{2}\varepsilon \).

Let \( \{x^k\} \) be a sequence generated by Algorithm 2. Then \( x^k \) converges to a point in \( P \) and \( (d_1^k, \ldots, d_N^k) \) converges to a point in \( D \).

\textbf{IV. Experiment}

In this section, we give a practical example of a simple problem that can be solved via Algorithm 2. Consider the constrained problem
\[ \min_{x \in [c, d]^n} \|b - Hx\|_2 + \mu\|x\|_1, \tag{16} \]
where \( b \in \mathbb{R}^n, \ c \in \mathbb{R}, \ d \in \mathbb{R}, \ \mu > 0, \ H = \frac{1}{\sqrt{n}} \tilde{H}, \) and \( \tilde{H} \in \mathbb{R}^{n \times n} \) is a lower-triangular matrix of ones. Griess and Lorenz studied a non-constrained, and therefore simpler, version of this problem in the context of inverse integration [14, Section 4.1]. Problem (16) can be solved via Algorithm 2 if we let \( \gamma > 0, \ \tau > 0 \) and make \( m = n, \ N = 1, \ L_1 = \text{Id}, \ r_1 = 0, \ z = 0, \) and, for all \( k, \ U_1^k = \gamma \text{Id}, \ U^k = \tau \text{Id}, \ e_1^k = 0, \ b_1^k = 0, \)
\[ \Lambda_1^k = \text{Id}, \ c_0 = 0, \ a_0 = 0, \ f = \|b - H : \|_2^2, \ g = \mu \|\cdot\|_1, \]
\[ h = \delta_{[c,d]^n} (\cdot, l_1 : u \rightarrow 0 \text{ if } u = 0, \ l_1 : u \rightarrow +\infty \text{ otherwise}. \]
If we take \( \Lambda^k \) to be a sequence of scalars, we recover a version of [18, Example 6.4]. However, inspired by the fast convergence properties of the methods discussed in Section II and following a similar reasoning to [14, Proposition 3.7], we consider the B-differential for the operator \( \text{prox}_{\mu\|\cdot\|_1} \) given in (10) and take \( \Lambda^k \) to be the inverse of
\[ \left( \text{prox}_{\mu\|\cdot\|_1} \right)^{-1} \left[ \begin{array}{cc} \tau [H^*_j I_j^k H^*_j I_j^k] & \tau [H^*_j I_j^k H^*_j A^k] \\ 0 & \text{Id} \end{array} \right] \]
where
\[ A^k \triangleq \{i \in \mathbb{N} : |x^k - 2\tau (H^* (Hx^k - b) + y_1^k)|_i \leq \mu \}, \]
\[ I^k \triangleq \{i \in \mathbb{N} : |x^k - 2\tau (H^* (Hx^k - b) + y_1^k)|_i > \mu \}, \]
\[ \text{and } \{P^k\} \text{ is a sequence of appropriate permutation matrices such that, given a vector } x^k, \text{ the first elements of the vector } P^k x \text{ correspond to the indices in } I^k \text{ and the last elements to the indices in } A^k \text{, for all } k. \]
By again following a similar reasoning to the one of [14, Section 3.3], it can be shown that Line 12 of Algorithm 2 can be rewritten in such a way that this algorithm is easily seen to be equivalent to an active-set method. In fact, that line is given by
\[ x^{k+1} \leftarrow P^k \left[ \begin{array}{cc} \left( [H_j^k I_j^k H_j^k I_j^k]^{-1} \right)^{-1} \\ 0 \end{array} \right] \]
where \[ e_x^k \triangleq \text{sgn} \left( x^k - 2\tau (H^* (Hx^k - b) + y_1^k) \right) \] and every \( k \). The dimension of the problem to solve at each iteration is given by the cardinality of the set \( I^k \). Naturally, the sparser the solution is estimated to be, the smaller the dimension of this problem is. In contrast, methods such as the alternating-direction method of multipliers (ADMM) [19] require the solution of a problem involving the full matrix \( H^* H \). This is the reason why semismooth Newton methods are able to achieve faster convergence rates in practice than others.

We simulate an example similar to the one studied by Griess and Lorenz [14, Section 4.1] but consider the noise to be Gaussian with a signal-to-noise ratio (SNR) of 30 dB. We have set \( \mu = 3 \times 10^{-3}, \ c = -80, \) and \( d = 52. \)
compared Algorithm 2 (denoted in what follows as Proposed) with ADMM and with the method by Condat [20] (CM) to solve (16). We manually tuned the different parameters of the three methods in order to achieve the fastest convergence results in practice. We arbitrarily chose the result of ADMM after $10^7$ iterations as representative of the solution given by the three methods. Fig. 1 illustrates the behavior of the three methods by showing the root-mean-squared error (RMSE) between the estimates of each method and the representative solution, as a function of time. The three methods were initialized with the zero vector. The experiments were performed using MATLAB on an Intel Core i7 CPU running at 3.20 GHz, with 32 GB of RAM.

In this example, we did not enforce assumptions (15) but verified in practice that they were satisfied. However, in more complex examples, it may be necessary to devise a strategy that generates a sequence $\{\Lambda^k\}$ satisfying these assumptions. This is akin to the necessity of devising globalization strategies in other Newton-like methods [13, Chapter 8].

### A. Appraisal

It is clear that, for this example, the proposed method has a much faster convergence than either CM or ADMM. This improvement in convergence is similar to the one observed in the methods discussed in Section II. In general, the sparser the solution is, the faster the method is as well. In order to benefit from this property, we must be able to solve the lower-dimensional linear system faster than the full system. This may not always be possible: for example, in problems that involve computations with the fast Fourier Transform (FFT) of a signal, we usually have only modest improvements in speed if we wish to compute only selected elements of the FFT.\(^2\) However, for large-scale problems and for highly-sparse signals, methods known as sparse FFTs [21] may be useful. We verified in other experiments not detailed here that the proposed method has a comparable convergence speed to ADMM in problems whose solutions are not sparse or where we cannot take advantage of their sparsity.

\(^2\)See http://www.fftw.org/pruned.html for details.

### V. Conclusions

In this work, we defined operator-weighted averaged operators, and showed that they can be used to construct a number of algorithms with good convergence properties. These algorithms have very broad applications, and seem to be particularly suitable to address problems with sparsity-inducing regularizers, as suggested by a simple experiment. Possible future directions to be explored are the possibility of relaxing the assumptions on $\Lambda^k$, and the study of which problems are most suitable to be tackled by these methods.

### References


