J. Math. Anal. Appl. 457 (2018) 1095–1117

Contents lists available at ScienceDirect

Journal of Mathematical Analysis and Applications

www.elsevier.com/locate/jmaa

Backward–forward algorithms for structured monotone inclusions in Hilbert spaces ☆

Hédy Attouch [a], Juan Peypouquet [b,*], Patrick Redont [a]

[a] *Institut Montpelliérain Alexander Grothendieck, IMAG UMR 5149 CNRS, Université Montpellier 2, place Eugène Bataillon, 34095 Montpellier Cedex 5, France*

[b] *Departamento de Matemática, Universidad Técnica Federico Santa María, Avenida España 1680, Valparaíso, Chile*

A R T I C L E I N F O

Article history:
Received 4 February 2016
Available online 16 June 2016
Submitted by H. Frankowska, A. Barbagallo, and G. Moscariello

Keywords:
Monotone inclusion
Forward–backward algorithm
Proximal-gradient method

A B S T R A C T

In this paper, we study the *backward–forward* algorithm as a splitting method to solve structured monotone inclusions, and convex minimization problems in Hilbert spaces. It has a natural link with the *forward–backward* algorithm and has the same computational complexity, since it involves the same basic blocks, but organized differently. Surprisingly enough, this kind of iteration arises when studying the time discretization of the regularized Newton method for maximally monotone operators. First, we show that these two methods enjoy remarkable involutive relations, which go far beyond the evident inversion of the order in which the forward and backward steps are applied. Next, we establish several convergence properties for both methods, some of which were unknown even for the forward–backward algorithm. This brings further insight into this well-known scheme. Finally, we specialize our results to structured convex minimization problems, the gradient-projection algorithms, and give a numerical illustration of theoretical interest.

1. Introduction

The forward–backward algorithm was introduced by Lions and Mercier [25] and Passty [34] in order to find a zero of the sum of two maximally monotone operators. It can be naturally traced back to the projected-gradient method considered by Goldstein [22] and Levitin and Polyak [24] for constrained optimization problems. Each iteration of the algorithm consists of a forward (explicit) step with respect to a cocoercive (thus Lipschitz-continuous) operator B, and a backward (implicit) step with respect to a general

☆ Effort sponsored by the Air Force Office of Scientific Research, USAF, under grant number FA9550-14-1-0056, FONDECYT Grant 1140829, Conicyt Anillo ACT-1106, ECOS-Conicyt Project C13E03, Millenium Nucleus ICM/FIC RC130003, Conicyt Project MATHAMSUD 15MATH-02, Conicyt Redes 140183, and Basal Project CMM Universidad de Chile.

* Corresponding author.
E-mail addresses: hedy.attouch@univ-montp2.fr (H. Attouch), juan.peypouquet@usm.cl (J. Peypouquet), patrick.redont@univ-montp2.fr (P. Redont).

http://dx.doi.org/10.1016/j.jmaa.2016.06.025
0022-247X/

maximally monotone operator A. A variant includes an additional relaxation step, which may improve its numerical performance.

Forward–backward algorithms have proved to be efficient tools for solving structured monotone inclusions, and convex minimization problems. They provide parallel splitting methods which can be easily implemented, and which are particularly interesting for large-scale systems. They play an important role in signal and image processing, especially when dealing with sparse optimization. They are also adapted to domain decomposition techniques for PDE's (see [7]).

An important number of contributions, dealing with various topics, have been devoted to the development of this flexible method. The forward–backward–forward algorithm deals with maximally monotone Lipschitz operators B that are not necessarily cocoercive (like linear skew-symmetric operators) with application to Lagrangian methods [17,13,41]. FISTA is an acceleration of the FB method based on Nesterov's approach [15,32,31] (see also [20] for applications to signal recovery). Approximate data and computational errors are considered in [40] among many others. An inertial forward–backward algorithm is studied in [10]. In [8,33], the method is coupled with approximation or penalization techniques. Based on Kurdyka–Lojasiewicz property, convergence of forward–backward algorithms has been recently obtained in a nonconvex, nonsmooth setting, for tame optimization and semi-algebraic problems [5,4,21,16]. For a recent account on these methods one can consult [6,13,15,19,40] and the bibliography therein.

To our knowledge, the existing methods of forward–backward type all consider the explicit step first and the implicit step next. The alternative, a backward–forward algorithm, has not been explored. Surprisingly enough, this kind of iteration arises when studying the time discretization of the regularized Newton method for maximally monotone operators proposed in [11], and thereafter extended to the case of structured monotone operators in [2]. A semi-implicit discretization of the dynamical system studied in [9] (different from the one considered in [10]) produces this type of methods as well.

Having in mind this connection with Newton-like systems, our original aim was to study backward–forward algorithms both theoretically and numerically, and assess their performance, especially in connection with traditional forward–backward methods. As research progressed, we found out some remarkable involutive relationships between the forward and backward steps. These properties allow us to understand forward–backward algorithms more deeply and obtain convergence results beyond the classical monotone setting. Moreover, we can account for an over-relaxed combination step and deduce further properties of the limits. When coupled with a relaxation step, which may accelerate convergence, the forward–backward and the backward–forward are different. Yet they share the same computational complexity (a gradient and a proximal operation) and the same convergence properties. They account for the numerical observation that reversing the order of the gradient and the proximal step is not important.

The paper is organized as follows: In Section 2, we describe the forward–backward and backward–forward algorithms, point out some relevant facts concerning set-valued operators, and present some *involutive* relations that allow to consider both algorithms in a somewhat unified manner. Convergence results – old and new – for both algorithms are presented in Section 3 in the operator setting. The case where the operators A and B derive from convex potentials is investigated in Section 4. A numerical illustration is given in Section 5, while further remarks and perspectives are commented in Section 6.

2. Forward–backward and backward–forward algorithms

Throughout this paper, $\mathscr{H}$ is a real Hilbert space with scalar product $\langle .,. \rangle$ and norm $\| \cdot \|$. We mostly adopt the definitions and notations of [13].

Let $F : \mathscr{H} \rightrightarrows \mathscr{H}$ be a set-valued operator. The inverse operator $F^{-1} : \mathscr{H} \rightrightarrows \mathscr{H}$ is defined by the relation $y \in F^{-1}x \Leftrightarrow x \in Fy$.

For $\gamma \in \mathbf{R} \setminus \{0\}$, the *resolvent of F of index γ* is the operator $J_{\gamma F} = (I + \gamma F)^{-1}$. For $\gamma \in \mathbf{R}$, the *Yosida approximation of F of index γ* is $F_\gamma = (F^{-1} + \gamma I)^{-1}$. In other words, $y \in F_\gamma x \Leftrightarrow y \in F(x - \gamma y)$. When