Open access • Journal Article • DOI:10.1137/0805023

Proximal decomposition on the graph of a maximal monotone operator

— Source link

Philippe Mahey, Said Oualibouch, Pham Dinh Tao

Institutions: Institut national des sciences appliquées de Rouen

Published on: 01 May 1995 - Siam Journal on Optimization (Society for Industrial and Applied Mathematics)

Topics: Strongly monotone, Pseudo-monotone operator and Inverse

Proximal Decomposition on the Graph of a Maximal Monotone Operator

Philippe Mahey, Said Oualibouch, Tao Pham Dinh

PROXIMAL DECOMPOSITION ON THE GRAPH OF A MAXIMAL MONOTONE OPERATOR*

PHILIPPE MAHEY[†], SAID OUALIBOUCH[‡], AND PHAM DINH TAO[§]

Abstract. We present an algorithm to solve: Find $(x, y) \in A \times A^{\perp}$ such that $y \in Tx$, where A is a subspace and T is a maximal monotone operator. The algorithm is based on the proximal decomposition on the graph of a monotone operator and we show how to recover Spingarn's decomposition method. We give a proof of convergence that does not use the concept of partial inverse and show how to choose a scaling factor to accelerate the convergence in the strongly monotone case. Numerical results performed on quadratic problems confirm the robust behaviour of the algorithm.

Key words. proximal point algorithm, partial inverse, convex programming

AMS subject classification. 90C25

1. Introduction.

We consider in this paper the following constrained inclusion problem: let X be a finite dimensional vector space and A a subspace of X. Let us denote by B the orthogonal subspace of A, i.e., $B = A^{\perp}$. Let T be a maximal monotone operator on X and denote its graph by $\mathrm{Gr}(T)$, i.e., $\mathrm{Gr}(T) = \{(x, y) \in X \times X | y \in Tx\}$. Then, the problem is to find $x \in A$ and $y \in B$ such that $y \in Tx$, which can be written:

$$\text{(P) Find } (x, y) \in X \times X \text{ such that } (x, y) \in A \times B \cap \mathrm{Gr}(T).$$

A typical situation, which is easily shown to give the form (P), is the problem of minimizing a convex lower semicontinuous function on a subspace. The particular applications we have in mind are the decomposition methods for separable convex programming. They have recently gained some new interest with the possibility of implementing them on massively parallel architectures to solve very large problems such as the ones that appear in network optimization or stochastic programming (see [1]). There are many different ways to transform a separable convex program in the form (P), but the general idea is to represent the coupling between the subsystems by a subspace of the product space of the copies of the primal and dual variables.

We are aiming here at the application of the Proximal Point Algorithm (PPA) (cf. [11]) to problem (P). In 1983, Spingarn [12] proposed a generalization of PPA to solve (P) that was based on the notion of the Partial Inverse operator. If we denote by x_A the orthogonal projection of x on a subspace A, the graph of the partial inverse operator T_A is given by

$$\mathrm{Gr}(T_A) = \{(x_A + y_B, y_A + x_B) \mid y \in Tx\}.$$

Applying the PPA to this operator leads to the Partial Inverse Method (PIM) which we summarize here.

ALGORITHM 1 (PIM). *At iteration k, $(x_k, y_k) \in A \times B$. Then, find (x'_k, y'_k) such that $x_k + y_k = x'_k + y'_k$ and $\frac{1}{c}(y'_k)_A + (y'_k)_B \in T((x'_k)_A + \frac{1}{c}(x'_k)_B)$.*

* Received by the editors March 26, 1993; accepted for publication (in revised form) February 14, 1994.

[†] ISIMA, B.P. 125, 63173 Aubière, Cedex, France (mahey@flamengo.isima.fr).

[‡] Institut d'Informatique et d'Intelligence Artificielle, Monruz 36, CH-2000 Neuchatel, Switzerland.

[§] LMAI, INSA Rouen, Place Emile Blondel, B.P. 8, 76131 Mont-Saint-Aignan, France.

Then, $(x_{k+1}, y_{k+1}) = ((x'_k)_A, (y'_k)_B)$.

The main problem that arises with this algorithm is the difficulty of performing the proximal step (1) when $c \neq 1$ in most interesting situations including the decomposition methods. When $c = 1$, then the proximal step is a proximal decomposition on the graph of T and the subspaces A and B only appear in the projection step. In §3 we present the resultant algorithm, indeed equivalent to PIM with $c = 1$. The convergence is proved without the need to consider the Partial Inverse operator. The iteration is now written in the following way.

Proximal decomposition. Find the unique (x'_k, y'_k) such that $x'_k + y'_k = x_k + y_k$ and $(x'_k, y'_k) \in \mathrm{Gr}(T)$ If $(x'_k, y'_k) \in A \times B$, then stop.
Else $(x_{k+1}, y_{k+1}) = ((x'_k)_A, (y'_k)_B)$.

The unique solution of the proximal decomposition step is given by

(1)
$$\begin{aligned} x'_k &= (I + T)^{-1}(x_k + y_k), \\ y'_k &= (I + T^{-1})^{-1}(x_k + y_k). \end{aligned}$$

Of course, only one proximal calculus is needed as $(I + T^{-1})^{-1} = I - (I + T)^{-1}$. We propose then a modified proximal decomposition algorithm by introducing scaling factors λ and μ. Indeed, problem (P) may be written in two ways :

$$\begin{aligned} y \in Tx &\iff x + \lambda y \in (I + \lambda T)x, \\ x \in T^{-1}y &\iff y + \mu x \in (I + \mu T^{-1})y, \end{aligned}$$

which induces the following fixed point iteration, a natural scaled version of (1).

Modified proximal decomposition.

(2)
$$\begin{aligned} x'_k &= (I + \lambda T)^{-1}(x_k + \lambda y_k), \\ y'_k &= (I + \mu T^{-1})^{-1}(y_k + \mu x_k). \end{aligned}$$

If $(x'_k, y'_k) \in A \times B$, then stop.
Else $(x_{k+1}, y_{k+1}) = ((x'_k)_A, (y'_k)_B)$.

It appears that the modified proximal step is uniquely determined and corresponds to a proximal decomposition on the graph of λT if $\lambda\mu = 1$. We recover then the scaled version of PIM proposed by Spingarn in [13]. It is mentioned in [6] that the performance of PIM is very sensitive to the scaling factor variations and we give an explanation of this fact, allowing its adjustment to an optimal value in the strongly monotone case.

In §4, we give some numerical results that confirm the accelerating properties of the scaling parameter.

2. The proximal decomposition on the graph of a maximal monotone operator. We recall here some known results on the "Prox" mapping $(I + T)^{-1}$ associated to a maximal monotone operator T and focus on the properties of the decomposition on the graph of T. More details on that subject can be found in [2] and [5].

Let T be a maximal monotone operator on a Hilbert space X. The graph of T, denoted by $\mathrm{Gr}(T)$, is defined by

$$\mathrm{Gr}(T) = \{(x, y) \in X \times X | y \in Tx\}.$$

Monotonicity implies that for all $x, x' \in X$ and for all $y \in Tx$, for all $y' \in Tx'$, $\langle y - y', x - x' \rangle \geq 0$. As T is maximal, its graph is not properly contained in the graph of any other monotone operator.

If T is strongly monotone, then there exists a positive ρ such that

$$\forall x, x' \in X \quad \text{and} \quad \forall y \in Tx, \quad \forall y' \in Tx', \quad \langle y - y', x - x' \rangle \geq \rho \|x - x'\|^2.$$

We say that the operator T is Lipschitz with constant L if

$$\forall x, x' \in X \quad \text{and} \quad \forall y \in Tx, \forall y' \in Tx', \|y - y'\| \leq L\|x - x'\|.$$

For monotone operators that share both properties, we get the following explicit bounds:

$$\rho\|x - x'\| \leq \|y - y'\| \leq L\|x - x'\|. \tag{3}$$

When T is a linear operator represented by a positive definite matrix $\mathcal{T}$, the best estimates for ρ and L are, respectively, the smallest and the largest eigenvalues of $\mathcal{T}$.

Of course, if T is maximal monotone, then for any $\lambda > 0$, λT is maximal monotone and if, moreover, T is strongly monotone with modulus ρ and Lipschitz with constant L, then λT is strongly monotone with modulus $\lambda\rho$ and Lipschitz with constant λL.

The resolvent associated with maximal monotone operator T is defined by $(I + T)^{-1}$. It is single-valued, defined on the whole space, and firmly nonexpansive, which means that, if we let $U = (I + T)^{-1}$ and $V = I - U$, then,

$$\forall x, x' \in X, \|Ux - Ux'\|^2 + \|Vx - Vx'\|^2 \leq \|x - x'\|^2 \tag{4}$$

or equivalently

$$\|Ux - Ux'\|^2 \leq \langle x - x', Ux - Ux' \rangle. \tag{5}$$

Related interesting facts on this characteristic property of resolvents may be found in theses by Martinet [9] and Eckstein [3] (see also [5]). Indeed, resolvents and maximal firmly nonexpansive mappings coincide and, following [7], one-to-one correspondences among these operators, maximal monotone, and maximal nonexpansive operators, may be stated. This fact is explored further in the appendix.

We introduce now the proximal decomposition on the graph of a maximal monotone operator.

Given a maximal monotone operator T and a vector $(x, y) \in X \times X$, there exists a unique pair $(u, v) \in X \times X$ called the proximal decomposition of (x, y) on the graph of T such that

$$u + v = x + y \quad \text{and} \quad (u, v) \in \mathrm{Gr}(T).$$

The unicity is a direct consequence of the maximality of T and we get

$$\begin{aligned} u &= (I + T)^{-1}(x + y), \\ v &= (I + T^{-1})^{-1}(x + y). \end{aligned}$$

3. The proximal decomposition algorithm. We return now to problem (P), which has been analyzed by Spingarn [13]. Let T be a maximal monotone operator on X. Let A be a subspace and B its orthogonal subspace. The problem is to find

$$(x,y) \in X \times X \text{ such that } (x,y) \in \ A \times B \cap \mathrm{Gr}(T).$$

This problem is a particular case of the general problem of finding a zero of the sum of two maximal monotone operators. The algorithms we are aiming at are splitting methods that alternate computations on each operator separately (see [8]). Indeed, most large-scale optimization problems can be formulated as the problem of minimizing a separable convex lower semicontinuous function on a very simple subspace which represents the coupling between the subsystems.

We propose then a generic algorithm that alternates a proximal decomposition on the graph of T with a projection on $A \times B$. Before going on with the analysis of the method, we observe that the other alternatives that come to mind to find a point in the intersection of two sets are not suitable.

1. We can use the classical successive projections method on the two sets. The problem is that $\mathrm{Gr}(T)$ is generally not convex in $X \times X$.

2. We cannot use another proximal decomposition on $A \times B$ (which is indeed the graph of the maximal monotone operator $\partial\chi_A$, the subdifferential of the indicator function of the set A), because it would lead back to the original point! Indeed, if $(x,y) \in A \times B$ and (u,v) is the proximal decomposition of $x+y$ on $\mathrm{Gr}(T)$, then $x = (u+v)_A$ and $y = (u+v)_B$, which means that (x,y) is the proximal decomposition of $u+v$ on the graph of $\partial\chi_A$.

The Algorithm PDG (proximal decomposition on the graph) is stated below.

ALGORITHM 2 (PDG). Let $(x_0,y_0) \in A \times B$. $k=0$.
If $(x_k,y_k) \in Gr(T)$, *then stop:* (x_k,y_k) *is a solution of* (P).
Else compute the proximal decomposition (u_k,v_k) of x_k+y_k on the graph of T. If $(u_k,v_k) \in A \times B$, *stop:* (u_k,v_k) is a solution of (P).
Else, $x_{k+1} = (u_k)_A$ and $y_{k+1} = (v_k)_B$.
$k = k+1$

An iteration of the algorithm may be formally stated as

$$(x,y) \in A \times B \mapsto \mathcal{L}(x,y) = x+y = z \in X \mapsto (u,v) = \mathcal{F}z \mapsto \mathcal{P}_{A\times B}(u,v) \in A \times B,$$

where L is isometric from $X \times X$ into X, $\mathcal{F}$ is the proximal decomposition operator from X into $X \times X$, and $P_{A\times B}$ is the projection on $A\times B$. Let us denote the composed mapping by

$$\mathcal{J} = \mathcal{P}_{A\times B} \circ \mathcal{F} \circ \mathcal{L}.$$

We verify now that any fixed point (x,y) of Algorithm PDG is a solution of (P). Indeed, $(x,y) = \mathcal{P}_{A\times B}(u,v)$ and $(u,v) = \mathcal{F}z$ with $z = x+y$. If $(u,v) \in A \times B$, then (x,y) is a solution of (P). Else, we have

$$(u-x, v-y) \in L = \{(a,b) \in X \times X | a+b = 0\}.$$

But, as $(x,y) = \mathcal{P}_{A\times B}(u,v)$, we can state

$$(u-x, v-y) \in B \times A.$$

A and B being orthogonal subspace, the unique intersection of L and $B \times A$ is $(0,0)$. Thus, $(x,y) = (u,v)$ and (x,y) solves (P).

On the other hand, if (x,y) is a solution of (P), $\mathcal{F}(x+y) = (x,y)$, and $(x,y) \in A \times B$, which means that (x,y) is a fixed point of Algorithm PDG.

The PDG Algorithm is a particular instance of Spingarn's Partial Inverse Method [12]. Indeed, when $c = 1$, the proximal step on the Partial Inverse operator T_A becomes: Find (x'_k, y'_k) such that : $x_k + y_k = x'_k + y'_k$ and $(y'_k)_A + (y'_k)_B \in T((x'_k)_A + (x'_k)_B)$, which means, of course, that (x'_k, y'_k) is the proximal decomposition of (x_k, y_k) on the graph of T. Thus, the convergence has been established by Spingarn who has used the properties of the PPA applied to the partial inverse operator. However, here we give a direct proof of this fact that does not use the concept of the Partial Inverse. The main interest is that we shall obtain as a corollary the numerical analysis of the scaled version of PDG in the strongly monotone case.

We prove first that the composed mapping $\mathcal{J}$ associated with Algorithm PDG is firmly nonexpansive. It can easily be seen that the mapping $\mathcal{U} = \mathcal{L} \circ \mathcal{J} \circ \mathcal{L}^{-1}$ is indeed the proximal operator associated to the Partial Inverse of T, i.e., $\mathcal{U} = (I + T_A)^{-1}$. But, we do not use this fact to prove that $\mathcal{J}$ is firmly nonexpansive.

THEOREM 3.1. *The mapping $\mathcal{J}$ associated to Algorithm* PDG *is firmly nonexpansive if and only if T is monotone. Moreover, it is defined on the whole space $A \times B$ if and only if T is maximal monotone.*

Proof. Let (x,y) and $(x',y') \in A \times B$, $z, z' \in \mathcal{L}(x,y), \mathcal{L}(x',y')$ respectively, i.e., $z = x + y$ and $z' = x' + y'$, $(u,v) \in \mathcal{F}(z)$ and $(u',v') \in \mathcal{F}(z')$, i.e., $u + v = z$, $u = (I+T)^{-1}z$ and $u' + v' = z'$, $u' = (I+T)^{-1}z'$. Finally, let (u_A, v_B) and $(u'_A, v'_B) \in A \times B$ be the respective projections of (u,v) and (u',v') on $A \times B$.

It is clear that, as $z \in (I+T)u$, $\mathrm{dom}(\mathcal{F}) = \mathrm{R}(I+T)$, and $\mathrm{dom}(\mathcal{J}) = \mathcal{L}^{-1}(\mathrm{dom}(\mathcal{F})) = \{(z_A, z_B) \in A \times B | z \in \mathrm{dom}(\mathcal{F})\}$.

Now, $\mathcal{J}$ is firmly nonexpansive if and only if

$$(6) \quad \begin{array}{l} \forall (x,y),(x',y') \in \mathrm{dom}(\mathcal{J}) \quad \text{and} \quad \forall (u_A, v_B) \in \mathcal{J}(x,y), (u'_A, v'_B) \in \mathcal{J}(x',y') \\ \langle (x,y) - (x',y'), (u_A, v_B) - (u'_A, v'_B) \rangle \geq \|(u_A, v_B) - (u'_A, v'_B)\|^2. \end{array}$$

But, we have

$$\langle (x,y) - (x',y'),.(u_A, v_B) - (u'_A, v'_B) \rangle = \langle x - x', (u - u')_A \rangle + \langle y - y', (v - v')_B \rangle$$

and, as $x, x' \in A$ and $y, y' \in B$

$$\begin{aligned} \langle x - x', (u - u')_A \rangle &= \langle z - z', (u - u')_A \rangle \\ &= \langle (u + v - u' - v', (u - u')_A \rangle \\ &= \langle (u - u') + (v - v'), (u - u')_A \rangle \end{aligned}$$

and

$$\begin{aligned} \langle y - y', (v - v')_B \rangle &= \langle z - z', (v - v')_B \rangle \\ &= \langle (u - u') + (v - v'), (v - v')_B \rangle. \end{aligned}$$

Hence, inequality (6) becomes

$$\begin{aligned} &\langle (u - u') + (v - v'), (u - u')_A \rangle \\ &\qquad + \langle (u - u') + (v - v'), (v - v')_B \rangle \geq \|(u - u')_A\|^2 + \|(v - v')_B\|^2. \end{aligned}$$

We can now use the orthogonal decomposition of $u - u'$ and $v - v'$ on the direct sum $A \oplus B$ to get

$$\begin{aligned}&\forall (u,v),(u',v') \in \mathrm{Gr}(T),\\ &\langle (u-u')_A,(v-v')_A\rangle + \langle (u-u')_B,(v-v')_B\rangle \geq 0.\end{aligned}$$

Finally, remarking that

$$\langle u-u', v-v'\rangle = \langle (u-u')_A,(v-v')_A\rangle + \langle (u-u')_B,(v-v')_B\rangle,$$

we can conclude that $\mathcal{J}$ is firmly nonexpansive if and only if T is monotone.

Moreover, as $\mathrm{dom}(\mathcal{J}) = \{(x,y) \in A \times B | x+y \in \mathrm{dom}(\mathcal{F})\}$, we obtain

$$\left.\begin{array}{l}\mathcal{J}\text{firmly nonexpansive}\\ \mathrm{dom}(\mathcal{J}) = A \times B\end{array}\right\} \Longleftrightarrow \left\{\begin{array}{l}T\text{monotone}\\ \mathrm{dom}(\mathcal{F}) = X\end{array}\right. \Longleftrightarrow T\text{maximal monotone.} \qquad \square$$

Assuming that (P) has a solution, the convergence of the algorithm follows directly from Opial's lemma (see [10]), which states that, if a fixed point exists, a firmly nonexpansive operator is asymptotically regular and generates a convergent sequence. This is the very same idea as used by Martinet in the original proof for the PPA [9] and developed further by Rockafellar who included approximate computations of the proximal steps [11] .

4. A scaled decomposition on the graph of T. We introduce now a scaled version of the decomposition on the graph of a maximal monotone operator.

DEFINITION 4.1. *Let $(x,y) \in X \times X$, T be a maximal monotone operator and λ a positive number. Then, the scaled proximal decomposition of (x,y) on the graph of T is the unique (u,v) such that*

$$\begin{aligned}u + \lambda v &= x + \lambda y,\\ (u,v) &\in \mathrm{Gr}(T).\end{aligned}$$

Again, the existence and unicity of that new decomposition is a consequence of T being maximal monotone. Indeed, if $v \in Tu$, we can write

$$\begin{aligned}u + \lambda v &\in u + \lambda T u\\ \Rightarrow u &= (I + \lambda T)^{-1}(u + \lambda v)\\ &= (I + \lambda T)^{-1}(x + \lambda y)\\ v &= \lambda^{-1}(x + \lambda y - u).\end{aligned}$$

Observe that we can also write the following inclusions using the inverse operator T^{-1} for a given positive μ:

$$\begin{aligned}&u \in T^{-1}v,\\ &v + \mu u \in v + \mu T^{-1} v,\\ &\text{then } v = (I + \mu T^{-1})^{-1}(v + \mu u).\end{aligned}$$

Now, if μ satisfies $\mu^{-1} = \lambda$, we get $v + \mu u = \mu(u + \lambda v)$ and, using the fact that $(\mu T)^{-1} z = T^{-1}(\mu^{-1} z)$, we obtain

$$\begin{aligned}v &= \lambda^{-1}(I + \mu T^{-1})^{-1}(u + \lambda v)\\ &= (I + \mu T^{-1})^{-1}(\mu x + y).\end{aligned}$$

Resuming, the scaled decomposition on the graph of T can be defined by

$$\begin{aligned} u &= (I+\lambda T)^{-1}(x+\lambda y), \\ v &= (I+\mu T^{-1})^{-1}(\mu x+y), \end{aligned} \tag{7}$$

which appears as a natural generalization of (1). But, in fact, only one scaling factor can be introduced to maintain the desired properties, this is why we must fix $\lambda\mu = 1$.

We can now describe the iteration of a scaled version of Algorithm PDG.

ALGORITHM 3 (SPDG). $(x_k, y_k) \in A \times B$.
Compute the scaled decomposition of (x_k, y_k) on the graph of T.

$$\begin{aligned} u_k &= (I+\lambda T)^{-1}(x_k+\lambda y_k), \\ v_k &= \lambda^{-1}(x_k+\lambda y_k - u_k). \end{aligned}$$

If $(u_k, v_k) \in A \times B$, then stop. Else, $x_{k+1} = (u_k)_A$ and $y_{k+1} = (v_k)_B$.

Observe that the scaled proximal decomposition can be stated in the following way.

Let $w = \lambda v$ and $r = \lambda y$. Then, if (u, v) is the scaled proximal decomposition of (x, y) on the graph of T, (u, w) is the proximal decomposition of (x, r) on the graph of λT. Hence, from the preceding section, we know that the sequence $\{(x_k, r_k)\}$ converges to a point in $A \times B \cap \mathrm{Gr}(\lambda T)$. This fact implies that the sequence $\{(x_k, y_k)\}$ converges to a solution of (P).

On the other hand, we can see that SPDG is equivalent to the scaled version of the Partial Inverse Method (with $c = 1$) described by Spingarn in [13, Algorithm 2, p. 208] for the minimization of a convex function on a subspace. It reduces, of course, to PDG, i.e., to PIM, when $\lambda = 1$. Again, as the decomposition on the graph of T is a proximal step, approximate rules for computations can be added as in [11] to get an implementable algorithm. We prefer to omit these details to focus on the accelerating properties of the scaling parameter, which constitute the main contribution of the present work.

To analyze the influence of the scaling parameter on the speed ratio of convergence of SPDG, we consider now the case where T is both strongly monotone and Lipschitz.

THEOREM 4.2. *When T is strongly monotone with modulus ρ and Lipschitz with constant L, then the convergence of the sequence $\{(x_k, r_k)\}$ generated by* SPDG *with $r_k = \lambda y_k$ is linear with speed ratio*

$$\sqrt{1 - \frac{2\lambda\rho}{(1+\lambda L)^2}}.$$

Proof. If $\mathcal{J}_\lambda$ is the composed operator associated to the monotone operator λT, we define as in Theorem 3.1 $(x, r), (x', r') \in A \times B$, $z = x + r$, $z' = x' + r'$, $(u, w), (u', w') \in \mathrm{Gr}(\lambda T)$. Then, $(u_A, w_B) \in \mathcal{J}_\lambda(x, r)$ and $(u'_A, w'_B) \in \mathcal{J}_\lambda(x', r')$.

The strong monotonicity of λT implies that

$$\forall w \in Tu, \quad w' \in Tu', \quad \langle w - w', u - u' \rangle \geq \lambda\rho \|u - u'\|^2 \tag{8}$$

and, as $z \in (I + \lambda T)u$ and $z' \in (I + \lambda T)u'$

$$\|z - z'\| \leq (1 + L)\|u - u'\|. \tag{9}$$

From the composed nature of $\mathcal{J}_\lambda$ and using the relations (8) and (9), we deduce the following bounds:

$$\begin{aligned}
\|(u_A, w_B) - (u'_A, w'_B)\|^2 &\leq \|u - u'\|^2 + \|w - w'\|^2 \\
&\leq \|z - z'\|^2 - 2\langle u - u', w - w'\rangle \\
&\leq \|z - z'\|^2 - 2\rho\|u - u'\|^2 \\
&\leq \left(1 - \frac{2\lambda\rho}{(1+\lambda L)^2}\right)\|z - z'\|^2 \\
&\leq \left(1 - \frac{2\lambda\rho}{(1+\lambda L)^2}\right)\|(x, r) - (x', r')\|^2.
\end{aligned}$$

Let (x^*, r^*) be the limit point of the sequence $\{(x_k, r_k)\}$. It is therefore a fixed point of the mapping $\mathcal{J}_\lambda$. Applying the above inequality to the pairs (x_{k+1}, r_{k+1}) and (x^*, r^*), we obtain the desired result:

$$\|(x_{k+1}, r_{k+1}) - (x^*, r^*)\|^2 \leq \left(1 - \frac{2\lambda\rho}{(1+\lambda L)^2}\right)\|(x_k, r_k) - (x^*, r^*)\|^2. \qquad \square$$

Observe that, as

$$L \geq \rho, r(\lambda) = \sqrt{1 - \frac{2\lambda\rho}{(1+\lambda L)^2}} < 1.$$

We easily deduce the theoretical optimal value for λ:

$$\bar{\lambda} = 1/L \quad \text{and} \quad r(\bar{\lambda}) = \sqrt{1 - \frac{\rho}{2L}}. \tag{10}$$

When T is a linear positive definite operator, we observe that bad conditioning implies a slowdown of the algorithm. The optimal value of the scaling parameter must be chosen very small if $L = \mu_{\max}$, the largest eigenvalue of the associated matrix, is very large. We may observe that the speed ratio obtained in Theorem 4.2 is the same as the one given in [8] for the Douglas–Rachford splitting algorithm. Indeed, the connection between that algorithm and the Partial Inverse Method has been established by Eckstein [3] and we give its precise meaning in the Appendix.

The influence of the Lipschitz constant on the number of iterations has been analyzed for quadratic convex functions that were minimized on a simple subspace. The sensitivity to that parameter is shown on the five graphics of Fig. 1 and 2 for different values of L, ρ, and the dimension of the space. These results are shown in Table 1. The influence of the scaling parameter on the number of iterations is illustrated by comparing columns iter($\bar{\lambda}$) (number of iterations when $\lambda = \bar{\lambda}$) and iter(1) (number of iterations when $\lambda = 1$). The number of iterations corresponds to the implementation of Algorithm PDG associated with the graph of λT. We show below why it is faster than the straightforward application of SPDG even if the primal sequences $\{x_k\}$ coincide in both algorithms.

It is also interesting to analyze the behaviour of the sequence $\{(x_k, y_k)\}$ and to look for some values of the scaling parameter such that, that sequence is mapped by a contraction. To be more precise, let $\mathcal{J}_\lambda$ and $\mathcal{H}_\lambda$ be the maps associated with the sequences $\{(x_k, r_k)\}$ and $\{(x_k, y_k)\}$, respectively. Then, if D_λ is the mapping defined by

$$D_\lambda(x, y) = (x, \lambda y),$$

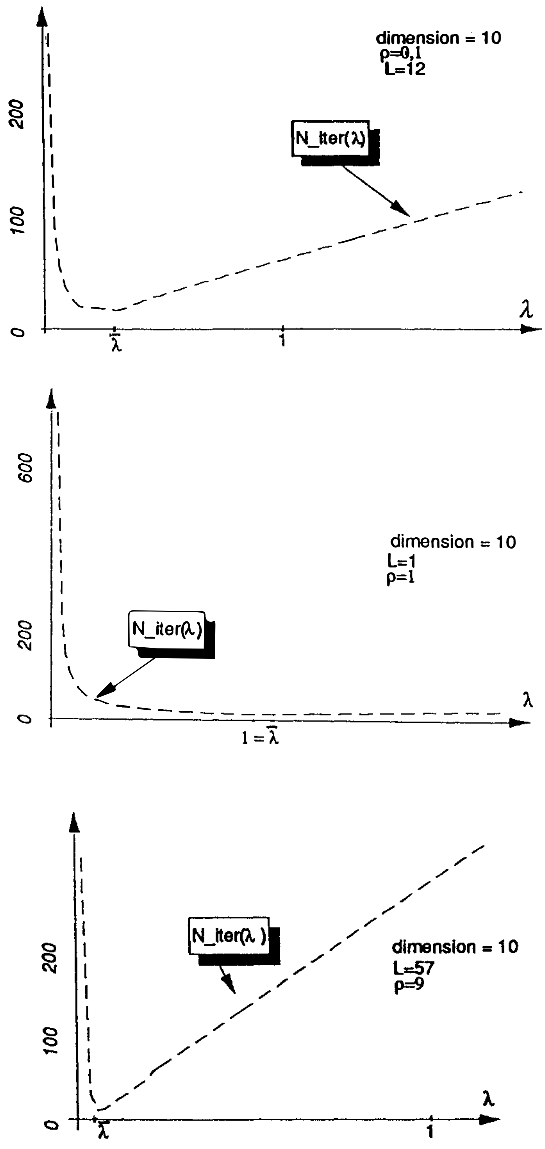

FIG. 1. *Number of iterations for* dim=10.

we can write the following correspondence:

$$\mathcal{H}_\lambda = D_\lambda^{-1} \circ \mathcal{J}_\lambda \circ D_\lambda.$$

As $(x_k, y_k) = D_\lambda^{-1}(x_k, r_k)$, we already know that the sequence $\{(x_k, y_k)\}$ converges when $\{(x_k, r_k)\}$ converges. Note that a direct proof of this fact seems rather hard to state. The reason is that $\mathcal{H}_\lambda$ is not necessarily a contractive map for any λ. We study below the conditions on λ to get a contraction in the strongly monotone case. In the strongly monotone and Lipschitz cases, we already know that $\mathcal{H}_\lambda$ is a contraction for $\lambda = 1$. The next theorem shows that this remains true if λ lies in a specific interval containing one.

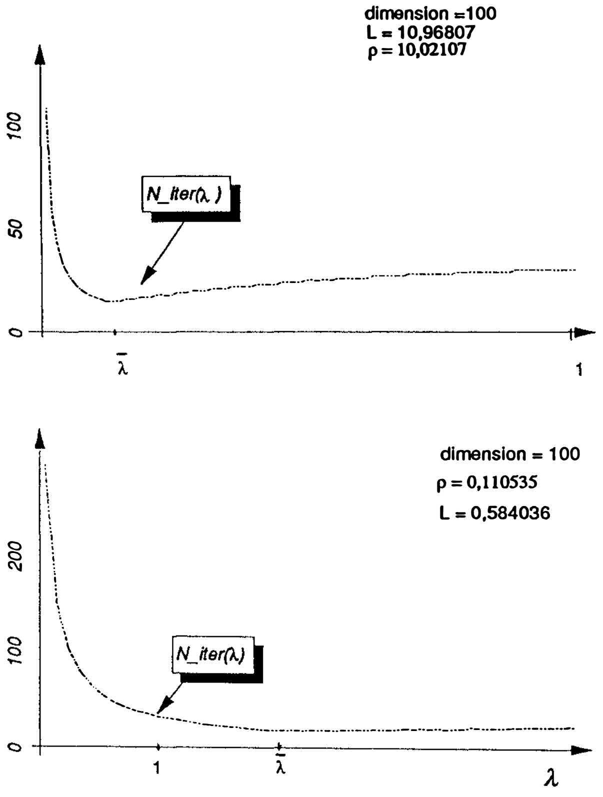

FIG. 2. *Number of iterations for* dim=100.

THEOREM 4.3. *Suppose that T is strongly monotone with modulus ρ and Lipschitz with constant L. Then, if $\lambda \in [1, \rho + \sqrt{1+\rho^2})$, the mapping $\mathcal{H}_\lambda$ is a contraction.*

Proof. Again let $u = (I + \lambda T)^{-1}(x + \lambda y)$ and $u' = (I + \lambda T)^{-1}(x' + \lambda y')$. We use successively the nonexpansiveness of the projection and the firmly nonexpansiveness of the resolvent to write

$$\begin{aligned}\|\mathcal{H}_\lambda(x,y) - \mathcal{H}_\lambda(x',y')\|_X^2 &\leq \|(u-u', \lambda^{-1}(x - x' + \lambda(y-y') - u + u'))\|_X^2 \\ &\leq \lambda^{-2}\|x - x'\|^2 + \|y - y'\|^2 + \frac{\lambda^2 - 2\lambda\rho - 1}{\lambda^2}\|u - u'\|^2.\end{aligned}$$

Using the Lipschitz property, we obtain

$$\|\mathcal{H}_\lambda(x,y) - \mathcal{H}_\lambda(x',y')\|_X^2 \leq \lambda^{-2}\left(1 + \frac{\lambda^2 - 2\lambda\rho - 1}{(1+\lambda L)^2}\right)(\|x - x'\|^2 + \lambda^2\|y - y'\|^2). \tag{11}$$

Hence, a sufficient condition that ensures that $\mathcal{H}_\lambda$ is a contraction is $\lambda \geq 1$ and $\theta(\lambda) = \lambda^2 - 2\lambda\rho - 1 < 0$. That condition does not depend on the Lipschitz constant (indeed, this happens because $0 < \rho < L$). We observe now that $\theta(1) = -2\rho < 0$ and the desired interval must be : $\lambda \in [1, \rho + \sqrt{1+\rho^2})$. □

The different behaviour of both sequences $\{(x_k, z_k)\}$ and $\{(x_k, y_k)\}$ is illustrated in Fig. 3. For a small λ, the second sequence (which is the one that will yield a solution

TABLE 1
Numerical tests for quadratic problems.

ρ	L	$1/L$	$\bar{\lambda}$	Iter($\bar{\lambda}$)	Iter(1)	dim	tolerance
0.1	0.584	1.712	2.05	17	34	100	0.01
	1.068	0.936	1.05	17	20		
	4.940	0.202	0.26	18	29		
	9.781	0.102	0.13	18	42		
	19.461	0.051	0.07	18	60		
	29.142	0.034	0.05	18	70		
	96.907	0.010	0.02	18	92		
1	1.968	0.508	0.58	17	19		
	5.840	0.171	0.12	18	35		
	10.681	0.094	0.12	17	44		
	20.361	0.049	0.06	18	60		
	30.042	0.033	0.05	18	70		
	49.404	0.020	0.03	18	82		
	97.807	0.010	0.02	19	92		
0.1	0.584	1.712	2.04	16	32	10	0.001
	1.068	0.936	1.21	16	18		
	4.940	0.202	0.241	17	27		
	9.781	0.102	0.121	17	39		
	19.461	0.051	0.061	17	54		
	29.142	0.034	0.041	17	63		
	96.907	0.010	0.021	17	75		
1	1.968	0.508	0.58	15	16		
	5.840	0.171	0.211	16	29		
	10.681	0.094	0.121	16	40		
	20.361	0.049	0.061	17	54		
	30.042	0.033	0.041	17	63		
	49.404	0.020	0.031	17	72		
	97.807	0.010	0.021	17	75		

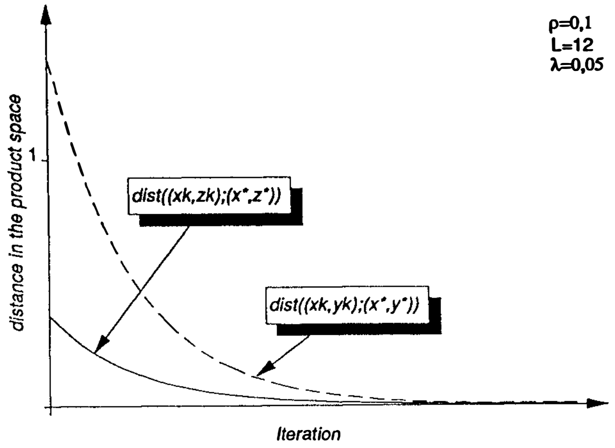

FIG. 3. *Comparison of both sequences*

for the original problem (P)) converges much slower even if it presents a monotonic decrease toward the fixed point.

We conclude with the following observations on the choice of the scaling parameter: if bad conditioning is due to a too-small ρ, then we must accelerate the convergence by choosing λ close to the optimal value $1/L$ (if it is not too far from

1!). If bad conditioning is due to a too-large L, then we may choose λ close to 1 in $[1, \rho + \sqrt{1+\rho^2})$.

Appendix. The relation between the partial inverse and the Douglas–Rachford splitting operator may be explained in the following way which is directly inspired by the work of Lawrence and Spingarn [7]. It was later derived by Eckstein and Bertsekas [4].

We recall the one-to-one correspondences among maximal monotone operators, maximal nonexpansive, and proximal operators as described in [7].

Let $\alpha : (x,y) \mapsto (x, 2y - x)$ be the one-to-one correspondence of the class of proximal operators onto the class of nonexpansive operators and let $\beta : (x,y) \mapsto (x+y, x-y)$ be the one-to-one correspondence of the class of monotone operators onto the class of nonexpansive operators. Following [7], let us define two types of composition operations.

Let $p_1 \star p_2 = \alpha^{-1}(\alpha(p_1) \circ \alpha(p_2))$ be the proximal operator obtained by composing two proximal operators p_1 and p_2 through their associated respective nonexpansive images (which give indeed another nonexpansive operator when composed). Likewise, let $T_1 \odot T_2 = \beta^{-1}(\beta(T_1) \circ \beta(T_2))$ be the monotone operator obtained by composing two monotone operators in the same way. A straightforward calculus shows that, if p_1 and p_2 are the resolvents of T_1 and T_2, respectively, then $p = p_1 \star p_2$ is the resolvent of $T = T_1 \odot T_2$. As observed in [7], we have the following interpretation of the $\star$ operation :

$$p_1 \star p_2 = p_1 \circ (2p_2 - I) + I - p_2,$$

which is the operator associated to the fixed point iteration of the Douglas–Rachford splitting method (see [8]). Observe that the nonexpansive operator $\alpha(p_1) \circ \alpha(p_2))$ is the operator associated with the Peaceman–Rachford iteration.

On the other side, it is shown in [7] that, when T_1 is the subdifferential mapping of the indicator function of a subspace A, i.e., $\mathrm{Gr}(T_1) = A \times A^{\perp}$, then $T_1 \odot T = T_A$, the Partial Inverse of T. Resuming these facts, we have the following proposition.

PROPOSITION. *Let T_1 and T_2 be two maximal monotone operators on X. The Douglas–Rachford splitting operator $p = p_1 \circ (2p_2 - I) + I - p_2$, where $p_1 = (I + \lambda T_1)^{-1}$ and $p_2 = (I + \lambda T_2)^{-1}$, is a proximal operator, indeed $p = (I+T)^{-1}$, where $T = \lambda T_1 \odot \lambda T_2$. Moreover, if $Gr(T_1) = A \times A^{\perp}$ and $T_2 = T$, then $p = (I + (\lambda T)_A)^{-1}$, the resolvent of the partial inverse of λT. Then, the Douglas–Rachford iteration applied to problem* (P) *is the partial inverse method associated to λT.* SPDG *is the corresponding algorithm defined in the product space $X \times X$.*

Observation. Clearly $(I + (\lambda T)_A)^{-1} \neq (I + \lambda T_A)^{-1}$. This point is crucial because the computation can only be performed in the first expression (this is then the SPDG Algorithm) or in the second expression with $\lambda = 1$.

REFERENCES

[1] D. BERTSEKAS AND J. TSITSIKLIS, *Parallel and Distributed Computation*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[2] H. BREZIS, *Opérateurs Maximaux Monotones*, Mathematics Studies 5, North Holland, Amsterdam, 1973.

[3] J. ECKSTEIN, *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, 1989.

[4] J. Eckstein and D. Bertsekas, *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.

[5] K. Goebel and W. Kirk, *Topics in Metric Fixed Point Theory*, Studies in Advanced Mathematics 28, Cambridge University Press, 1990.

[6] H. Idrissi, O. Lefebvre, and C. Michelot, *Applications and numerical convergence of the partial inverse method*, in Optimization, Lecture Notes in Math. 1405, S. Dolecki, ed., Springer-Verlag, 1989, New York, pp. 39–54.

[7] J. Lawrence and J. Spingarn, *On fixed points of nonexpansive piecewise isometric mappings*, Proc. London Math. Soc., 55 (1987), pp. 605–624.

[8] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.

[9] B. Martinet, *Algorithmes pour la Résolution de Problèmes d' Optimisation et de Minimax*, Thèse d'Etat, University de Grenoble, 1972.

[10] Z. Opial, *Weak convergence of the successive approximations for nonexpansive mappings in Banach spaces*, Bull. Amer. Math. Soc., 73 (1967), pp. 591–597.

[11] R. Rockafellar, *Monotone operators and the proximal point algorithm in convex programming*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[12] J. Spingarn, *Partial inverse of a monotone operator*, Appl. Math. Optim., 10 (1983), pp. 247–265.

[13] ———, *Applications of the method of partial inverse to convex programming: Decomposition*, Math. Programming, 32 (1985), pp. 199–223.