Mathematical Programming manuscript No.
(will be inserted by the editor)

On the equivalence of the primal-dual hybrid gradient method and Douglas–Rachford splitting

Daniel O'Connor · Lieven Vandenberghe

Abstract The primal-dual hybrid gradient (PDHG) algorithm proposed by Esser, Zhang, and Chan, and by Pock, Cremers, Bischof, and Chambolle is known to include as a special case the Douglas–Rachford splitting algorithm for minimizing the sum of two convex functions. We show that, conversely, the PDHG algorithm can be viewed as a special case of the Douglas–Rachford splitting algorithm.

Keywords Douglas–Rachford splitting · primal-dual algorithms · monotone operators · proximal algorithms

Mathematics Subject Classification (2000) 47N10 · 49M27 · 49M29 · 65K05 · 90C25

1 Introduction

The primal-dual first order algorithm discussed in [4, 15, 27] applies to convex optimization problems in the form

$$\text{minimize} \quad f(x) + g(Ax), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are closed convex functions with nonempty domains, and A is an $m \times n$ matrix. It uses the iteration

$$\bar{x}^k = \text{prox}_{\tau f}(x^{k-1} - \tau A^T z^{k-1}) \tag{2a}$$
$$\bar{z}^k = \text{prox}_{\sigma g^*}(z^{k-1} + \sigma A(2\bar{x}^k - x^{k-1})) \tag{2b}$$
$$x^k = x^{k-1} + \rho_k(\bar{x}^k - x^{k-1}) \tag{2c}$$
$$z^k = z^{k-1} + \rho_k(\bar{z}^k - z^{k-1}). \tag{2d}$$

The function g^* in (2b) is the conjugate of g, and $\text{prox}_{\tau f}$ and $\text{prox}_{\sigma g^*}$ are the proximal operators of τf and σg^* (see §2). The parameters σ and τ are positive constants that satisfy $\sigma\tau\|A\|^2 \le 1$, where $\|A\|$ is the 2-norm (spectral norm) of A, and ρ_k is a relaxation parameter in $(0, 2)$ that can change in each iteration if it is bounded away from 0 and 2, *i.e.*, $\rho_k \in [\epsilon, 2 - \epsilon]$ for some $\epsilon > 0$. With these parameter values, the iterates x^k, z^k converge to a solution of the optimality conditions

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix}, \tag{3}$$

if a solution exists [11, Theorem 3.3.]. Following the terminology in the survey paper [5, page 212], we use the name *primal-dual hybrid gradient* (PDHG) algorithm for the algorithm (2). The interested reader can find a discussion of the origin of this name in [5, footnote 11].

D. O'Connor
Department of Mathematics and Statistics, University of San Francisco
E-mail: daniel.v.oconnor@gmail.com

L. Vandenberghe
Department of Electrical and Computer Engineering, University of California Los Angeles
E-mail: vandenbe@ucla.edu

An older method is Lions and Mercier's Douglas–Rachford splitting (DRS) method for minimizing the sum of two closed convex functions $f(x) + g(x)$, by solving the optimality condition

$$0 \in \partial f(x) + \partial g(x),$$

see [8, 14, 20] [1, §2.7]. The DRS method with relaxation uses the iteration

$$\bar{x}^k = \mathrm{prox}_{tf}(y^{k-1}) \tag{4a}$$
$$y^k = y^{k-1} + \rho_k(\mathrm{prox}_{tg}(2\bar{x}^k - y^{k-1}) - \bar{x}^k), \tag{4b}$$

where t is a positive parameter and $\rho_k \in (0, 2)$. Chambolle and Pock [4, §4.2] and Condat [11, p.467] observe that this is a special case of the PDHG algorithm. Although they discuss the connection only for $\rho_k = 1$, it is easily seen to hold in general. We take $A = I$, $\tau = t$, $\sigma = 1/t$ in algorithm (2):

$$\begin{aligned}
\bar{x}^k &= \mathrm{prox}_{tf}(x^{k-1} - tz^{k-1}) \\
\bar{z}^k &= \mathrm{prox}_{t^{-1}g^*}(z^{k-1} + \frac{1}{t}(2\bar{x}^k - x^{k-1})) \\
x^k &= (1 - \rho_k)x^{k-1} + \rho_k\bar{x}^k \\
z^k &= (1 - \rho_k)z^{k-1} + \rho_k\bar{z}^k.
\end{aligned}$$

Next we replace z^k by a new variable $y^k = x^k - tz^k$:

$$\bar{x}^k = \mathrm{prox}_{tf}(y^{k-1}) \tag{5a}$$
$$\bar{z}^k = \mathrm{prox}_{t^{-1}g^*}(\frac{1}{t}(2\bar{x}^k - y^{k-1})) \tag{5b}$$
$$x^k = (1 - \rho_k)x^{k-1} + \rho_k\bar{x}^k \tag{5c}$$
$$y^k = x^k - (1 - \rho_k)(x^{k-1} - y^{k-1}) - \rho_k t\bar{z}^k. \tag{5d}$$

Substituting the expressions for $\bar{z}^k$ and x^k from lines (5b) and (5c) in (5d) gives

$$\begin{aligned}
\bar{x}^k &= \mathrm{prox}_{tf}(y^{k-1}) \\
y^k &= y^{k-1} + \rho_k(\bar{x}^k - y^{k-1} - t\mathrm{prox}_{t^{-1}g^*}(\frac{1}{t}(2\bar{x}^k - y^{k-1}))).
\end{aligned}$$

If we now use the Moreau identity $t\mathrm{prox}_{t^{-1}g^*}(u/t) = u - \mathrm{prox}_{tg}(u)$, we obtain the DRS iteration (4). Hence, the DRS algorithm is the PDHG algorithm with $A = I$ and $\sigma = 1/\tau$. The purpose of this note is to point out that, in turn, the PDHG algorithm can be derived from the DRS algorithm. As we will see in §4, the PDHG algorithm coincides with the DRS algorithm applied to a reformulation of (1). This result has interesting implications for the analysis of the PDHG algorithm. It allows us to translate known convergence results from the literature on DRS to the PDHG algorithm. It also simplifies the development and analysis of extensions of the PDHG algorithm, such as the three-operator extensions we will discuss in §6. Finally, since the reformulation used to show the equivalence does not require duality, it immediately leads to a primal form of the PDHG algorithm, which is better suited for non-convex extensions than the primal-dual form (2). This is disucssed in §5.

The rest of the paper is organized as follows. We start with a short review of monotone operators and the DRS algorithm (Sections 2 and 3). Section 4 contains the main result of the paper, the derivation of the PDHG algorithm from the DRS algorithm. In Sections 5–7 we discuss some variations and extensions of the results in Section 4.

2 Background material

2.1 Monotone operators

A *set valued operator* on $\mathbb{R}^n$ assigns to every $x \in \mathbb{R}^n$ a (possibly empty) subset of $\mathbb{R}^n$. The image of x under the operator F is denoted $F(x)$. If $F(x)$ is a singleton we usually write $F(x) = y$ instead of $F(x) = \{y\}$. If F is linear, $F(x) = Ax$, we do not distinguish between the operator F and the matrix A.

In particular, the symbol I is used both for the identity matrix and for the identity operator $F(x) = x$. The graph of an operator F is the set

$$\mathrm{gr}(F) = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^n \mid y \in F(x)\}.$$

The operator F is *monotone* if

$$(y - \hat{y})^T (x - \hat{x}) \geq 0 \qquad \forall (x, y), (\hat{x}, \hat{y}) \in \mathrm{gr}(F). \tag{6}$$

A monotone operator is *maximal monotone* if its graph is not contained in the graph of another monotone operator.

The inverse operator $F^{-1}(x) = \{y \mid x \in F(y)\}$ of a maximal monotone operator F is maximal monotone. We also define left and right scalar multiplications as

$$(\lambda F)(x) = \{\lambda y \mid y \in F(x)\}, \qquad (F\mu)(x) = \{y \mid y \in F(\mu x)\}.$$

Since $(\lambda F)\mu = \lambda(F\mu)$ we can write this operator as $\lambda F \mu$. If F is maximal monotone and $\lambda\mu > 0$, then $\lambda F \mu$ is maximal monotone. The inverse operation and the scalar multiplications are linear operations on the graphs:

$$\mathrm{gr}(F^{-1}) = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \mathrm{gr}(F), \qquad \mathrm{gr}(\lambda F \mu) = \begin{bmatrix} \mu^{-1} I & 0 \\ 0 & \lambda I \end{bmatrix} \mathrm{gr}(F).$$

From this, one easily verifies that $(\lambda F \mu)^{-1} = \mu^{-1} F^{-1} \lambda^{-1}$.

Maximal monotone operators are important in convex optimization because the subdifferential ∂f of a closed convex function is maximal monotone. Its inverse is the subdifferential of the conjugate $f^*(y) = \sup_x (y^T x - f(x))$ of f:

$$(\partial f)^{-1} = \partial f^*.$$

2.2 Resolvents, reflected resolvents, and proximal operators

The operator $J_F = (I + F)^{-1}$ is known as the *resolvent* of F. The value $J_F(x)$ of the resolvent at x is the set of all vectors y that satisfy

$$x - y \in F(y). \tag{7}$$

A fundamental result states that the resolvent of a maximal monotone operator is single-valued and has full domain, *i.e.*, the equation (7) has a unique solution for every x [14, Theorem 2]. The operator $R_F = 2J_F - I$ is called the *reflected resolvent* of F. The graphs of the resolvent and the reflected resolvent are related to the graph of F by simple linear mappings:

$$\mathrm{gr}(F) = \begin{bmatrix} 0 & I \\ I & -I \end{bmatrix} \mathrm{gr}(J_F) = \frac{1}{2} \begin{bmatrix} I & I \\ I & -I \end{bmatrix} \mathrm{gr}(R_F). \tag{8}$$

The resolvent of the subdifferential of a closed convex function f is called the *proximal operator* of f, and denoted $\mathrm{prox}_f = J_{\partial f}$. The defining equation (7) is the optimality condition of the optimization problem in

$$\mathrm{prox}_f(x) = \underset{y}{\mathrm{argmin}}\,(f(y) + \frac{1}{2}\|y - x\|^2), \tag{9}$$

where $\| \cdot \|$ is the Euclidean norm.

The following calculus rules can be verified from the definition (see also [1, Chapter 23] and, for proximal operators, [9, Table 10.1]). We assume F is maximal monotone and f is closed and convex, and that $\lambda > 0$.

1. *Right scalar multiplication.* The resolvent and reflected resolvent of $F\lambda$ are given by

$$J_{F\lambda}(x) = \lambda^{-1} J_{\lambda F}(\lambda x), \qquad R_{F\lambda}(x) = \lambda^{-1} R_{\lambda F}(\lambda x). \tag{10}$$

The proximal operator of the function $g(x) = f(\lambda x)$ is

$$\mathrm{prox}_g(x) = \frac{1}{\lambda} \mathrm{prox}_{\lambda^2 f}(\lambda x). \tag{11}$$

2. *Inverse.* The resolvents of F and its inverse satisfy the identities

$$J_F(x) + J_{F^{-1}}(x) = x, \qquad R_F(x) + R_{F^{-1}}(x) = 0 \tag{12}$$

for all x. Using $(\lambda F)^{-1} = F^{-1}\lambda^{-1}$ and the previous property, we also have

$$J_{\lambda F}(x) + \lambda J_{\lambda^{-1}F^{-1}}(x/\lambda) = x, \qquad R_{\lambda F}(x) + \lambda R_{\lambda^{-1}F^{-1}}(x/\lambda) = 0. \tag{13}$$

Applied to a subdifferential $F = \partial f$, the first identity in (13) is known as the *Moreau identity:*

$$\mathrm{prox}_{\lambda f}(x) + \lambda \mathrm{prox}_{\lambda^{-1}f^*}(x/\lambda) = x.$$

3. *Composition with linear mapping.* Assume A is an $n \times m$ matrix with $AA^T = I$ and F is a maximal monotone operator on $\mathbb{R}^m$. Then the operator $G(x) = A^T F(Ax)$ is maximal monotone, and the resolvents of G and G^{-1} are given by

$$J_G(x) = (I - A^T A)x + A^T J_F(Ax), \qquad J_{G^{-1}}(x) = A^T J_{F^{-1}}(Ax). \tag{14}$$

Combining (14) with the scaling rule (10), we find formulas for the resolvents of the operator $G(x) = A^T F(Ax)$ and its inverse, if $AA^T = \mu I$ and $\mu > 0$:

$$J_G(x) = \frac{1}{\mu}\left((\mu I - A^T A)x + A^T J_{\mu F}(Ax)\right), \qquad J_{G^{-1}}(x) = \frac{1}{\mu} A^T J_{(\mu F)^{-1}}(Ax), \tag{15}$$

and

$$R_G(x) = -R_{G^{-1}}(x) = \frac{1}{\mu}\left((\mu I - A^T A)x + A^T R_{\mu F}(Ax)\right).$$

It is useful to note that (10) and the identities for J_G in (14) and (15) hold for general (possibly nonmonotone) operators F. For (10) this is obvious from the definitions of the resolvent and scalar-operator multiplication. Since it will be important in §5, we prove the first equality in (14) for a general operator F. Suppose $y \in J_G(x)$, *i.e.*,

$$x - y \in A^T F(Ay). \tag{16}$$

Every vector y can be decomposed as $y = \hat{y} + A^T v$, where $A\hat{y} = 0$. If we make this substitution in (16) and use $AA^T = I$, we get

$$x - A^T v - \hat{y} \in A^T F(v). \tag{17}$$

Multiplying with A on both sides shows that v satisfies $Ax - v \in F(v)$. By definition of the resolvent this means that $v \in J_F(Ax)$. Multiplying both sides of (17) with $I - A^T A$ shows $\hat{y} = (I - A^T A)x$. Putting the two components together we find that

$$y = (I - A^T A)x + A^T v \in (I - A^T A)x + A^T J_F(Ax).$$

We conclude that $J_G(x) \subseteq (I - A^T A)x + A^T J_F(Ax)$. Conversely, suppose $v \in J_F(Ax)$, *i.e.*, $Ax - v \in F(v)$. Define $y = (I - A^T A)x + A^T v$. Then $Ay = v$ and

$$x - y = A^T(Ax - v) \in A^T F(v) = A^T F(Ay).$$

This shows that $y \in J_G(x)$, so $(I - A^T A)x + A^T J_F(Ax) \subseteq J_G(x)$.

The identities (12) and (13), on the other hand, are not necessarily valid for general operators. Their proofs depend on maximal monotonicity of F and, in particular, the fact that the resolvents $J_F(x)$ and $J_{F^{-1}}(x)$ are singletons for every x. However, if the identities in (12) are written as $J_{F^{-1}} = I - J_F$ and $R_F = -R_{F^{-1}}$, they do hold for general operators [30, Lemma 12.15].

μ-strong monotonicity	β-Lipschitz continuity	$1/\beta$-cocoercivity
$M = \begin{bmatrix} -2\mu & 1 \\ 1 & 0 \end{bmatrix}$	$M = \begin{bmatrix} \beta^2 & 0 \\ 0 & -1 \end{bmatrix}$	$M = \begin{bmatrix} 0 & \beta \\ \beta & -2 \end{bmatrix}$

Table 1 Each of the three properties is equivalent to a quadratic inequality (18) for the matrix M shown in the table. The parameters must satisfy $\mu > 0$, $\beta > 0$.

2.3 Strong monotonicity and cocoercivity

The definition of monotonicity (6) can be written as

$$\begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix}^T \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \geq 0 \qquad \forall (x, y), (\hat{x}, \hat{y}) \in \operatorname{gr}(F).$$

Several other important properties of operators can be expressed as differential quadratic forms on the graph. This is summarized in Table 1. Each of the three properties in the table is defined as

$$\begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix}^T \begin{bmatrix} M_{11}I & M_{12}I \\ M_{21}I & M_{22}I \end{bmatrix} \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \geq 0 \qquad \forall (x, y), (\hat{x}, \hat{y}) \in \operatorname{gr}(F) \tag{18}$$

where M is the 2×2-matrix given in the table. Strong convexity with parameter μ is equivalent to monotonicity of the operator $F - \mu I$. If an operator F is $(1/\beta)$-cocoercive, then it is β-Lipschitz continuous, as is easily seen, for example, from the matrix inequality

$$\begin{bmatrix} \beta^2 & 0 \\ 0 & -1 \end{bmatrix} \succeq \begin{bmatrix} 0 & \beta \\ \beta & -2 \end{bmatrix}.$$

The converse is not true in general, but does hold in the important case when F is the subdifferential of a closed convex function with full domain. In that case, $(1/\beta)$-cocoercivity and β-Lipschitz continuity are both equivalent to the property that $\beta I - F$ is monotone.

The resolvent of a monotone operator is 1-cocoercive (or *firmly nonexpansive*), and its reflected resolvent is 1-Lipschitz continuous, *i.e.*, nonexpansive. These facts follow from (8) and

$$\begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix},$$

respectively,

$$\frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^T \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

3 Douglas–Rachford splitting algorithm

The DRS algorithm proposed by Lions and Mercier [20] is an algorithm for finding zeros of a sum of two maximal monotone operators, *i.e.*, solving

$$0 \in F(x) + G(x). \tag{19}$$

It can be used to minimize the sum of two closed convex functions f and g by taking $F = \partial f$ and $G = \partial g$. The basic version of the method is the averaged fixed-point iteration

$$y^k = \frac{1}{2} y^{k-1} + \frac{1}{2} R_{tG}(R_{tF}(y^{k-1})) \tag{20}$$

where t is a positive constant. (This concise formulation of the DRS iteration in terms of the reflected resolvents is not used in the original paper [20] and appears in [2].) The right-hand side of (20) can be evaluated in three steps:

$$x^k = J_{tF}(y^{k-1}) \tag{21a}$$

$$w^k = J_{tG}(2x^k - y^{k-1}) \tag{21b}$$

$$y^k = y^{k-1} + w^k - x^k. \tag{21c}$$

It can be shown that y is a fixed point ($y = R_{tG}(R_{tF}(y))$) if and only if $x = J_{tF}(y)$ satisfies (19). In the following sections we discuss two extensions that have been proposed to speed up the method.

3.1 Relaxation

The relaxed DRS algorithm uses the iteration $y^k = T_k(y^{k-1})$ with

$$T_k(y) = (1-\alpha_k)y + \alpha_k R_{tG}(R_{tF}(y)). \tag{22}$$

The algorithm parameter t is a positive constant and $\alpha_k \in (0,1)$. The iteration reduces to (20) if $\alpha_k = 1/2$. The right-hand side of (22) can be calculated in three steps:

$$x^k = J_{tF}(y^{k-1}) \tag{23a}$$
$$w^k = J_{tG}(2x^k - y^{k-1}) \tag{23b}$$
$$y^k = y^{k-1} + \rho_k(w^k - x^k) \tag{23c}$$

where $\rho_k = 2\alpha_k$. Substituting $J_{tF} = \mathrm{prox}_{tf}$ and $J_{tG} = \mathrm{prox}_{tg}$ gives the version of the algorithm presented in (4). Using the relation between J_{tG} and $J_{(tG)^{-1}}$ we can write the algorithm equivalently as

$$x^k = J_{tF}(y^{k-1}) \tag{24a}$$
$$z^k = J_{(tG)^{-1}}(2x^k - y^{k-1}) \tag{24b}$$
$$y^k = (1-\rho_k)y^{k-1} + \rho_k(x^k - z^k). \tag{24c}$$

Fundamental references on the convergence of the DRS algorithm include [1, 7, 8, 10, 14, 20]. The weakest conditions for convergence are maximal monotonicity of the operators F and G, and existence of a solution. Convergence follows from the fact that $R_{tG} \circ R_{tF}$ is the composition of two nonexpansive operators, and therefore nonexpansive itself.

Convergence results that allow for errors in the evaluation of the resolvents are given in [14, Theorem 7], [7, Corollary 5.2]. If we add error terms d_k and e_k in (23) and write the inexact algorithm as

$$x^k = J_{tF}(y^{k-1}) + d_k \tag{25a}$$
$$w^k = J_{tG}(2x^k - y^{k-1}) + e_k \tag{25b}$$
$$y^k = y^{k-1} + \rho_k(w^k - x^k), \tag{25c}$$

then [7, Corollary 5.2] implies the following. Suppose the inclusion problem $0 \in F(x) + G(x)$ has a solution. If t is a positive constant, $\rho_k \in (0,2)$ for all k, and

$$\sum_{k\geq 0} \rho_k(\|d_k\| + \|e_k\|) < \infty, \qquad \sum_{k\geq 0} \rho_k(2-\rho_k) = \infty,$$

then the sequences w^k, x^k, y^k converge, and the limit of x^k is a solution of (19).

Under additional assumptions, linear rates of convergence obtain; see [13, 17, 18, 20] and the overview of the extensive recent literature on this subject in [17]. We mention the following result from [17]. Suppose F is μ-strongly monotone and G is $(1/\beta)$-cocoercive. Then the operator T_k defined in (22) is η_k-Lipschitz continuous with

$$\eta_k = |1 - 2\alpha_k + \alpha_k\kappa| + \alpha_k\kappa, \qquad \kappa = \frac{\beta t + 1/(\mu t)}{1 + \beta t + 1/(\mu t)}.$$

This is a contraction ($\eta_k < 1$) if $\alpha_k \in (0,1)$. The contraction factor η_k is minimized by taking $t = 1/\sqrt{\beta\mu}$ and

$$\alpha_k = \frac{1}{2-\kappa} = \frac{2+\sqrt{\mu/\beta}}{2(1+\sqrt{\mu/\beta})},$$

and its minimum value is

$$\eta = \frac{1}{1+\sqrt{\mu/\beta}}.$$

If T_k is η-Lipschitz continuous with $\eta < 1$, then the convergence of y^k is R-linear, *i.e.*, $\|y^k - y^\star\| \leq \eta^k\|y^0 - y^\star\|$. Since J_{tF} is nonexpansive, $x^k = J_{tF}(y^k)$ converges at the same rate.

3.2 Acceleration

Suppose F is μ-strongly monotone. The following extension of the DRS algorithm (21) is Algorithm 2 in [12] applied to the monotone inclusion problem (19):

$$\begin{aligned} x^k &= J_{t_k F}(y^{k-1}) && (26a)\\ w^k &= J_{t_{k+1} G}((1+\theta_k)x^k - \theta_k y^{k-1}) && (26b)\\ y^k &= \theta_k y^{k-1} + w^k - \theta_k x^k. && (26c) \end{aligned}$$

The parameters θ_k and t_k are defined recursively as

$$t_{k+1} = \theta_k t_k, \qquad \theta_k = \frac{1}{\sqrt{1+2\mu t_k}}, \qquad k = 1, 2, \ldots,$$

starting at some positive t_1. If we define $z^k = (1/t_{k+1})(x^k - y^k)$ and use the identity (13), we can write the algorithm as

$$\begin{aligned} x^k &= J_{t_k F}(x^{k-1} - t_k z^{k-1}) && (27a)\\ z^k &= J_{t_{k+1}^{-1} G^{-1}}(z^{k-1} + \frac{1}{t_{k+1}}(x^k + \theta_k(x^k - x^{k-1}))). && (27b) \end{aligned}$$

Davis and Yin [12, Theorem 1.2] show that if $x^\star$ is the solution of $0 \in F(x) + G(x)$ (the solution is necessarily unique, because F is assumed to be strongly monotone), then

$$\|x^k - x^\star\|^2 = O(\frac{1}{k^2}).$$

4 PDHG from DRS

We now consider the problem of finding a solution x of the inclusion problem

$$0 \in F(x) + A^T G(Ax), \tag{28}$$

where F is a maximal monotone operator on $\mathbb{R}^n$, G is a maximal monotone operator on $\mathbb{R}^m$, and A is an $m \times n$ matrix. For $F = \partial f$, $G = \partial g$, this is the optimality condition for the optimization problem (1).

To derive the PDHG algorithm from the DRS algorithm, we reformulate (28) as follows. Choose a positive γ that satisfies $\gamma\|A\| \leq 1$ and any matrix $C \in \mathbb{R}^{m\times p}$, with $p \geq m - \mathbf{rank}(A)$, such that the matrix

$$B = \begin{bmatrix} A & C \end{bmatrix} \tag{29}$$

satisfies $BB^T = \gamma^{-2} I$ (for example, the $m \times m$ matrix $C = (\gamma^{-2}I - AA^T)^{1/2}$). Problem (28) is equivalent to

$$0 \in \tilde{F}(u) + \tilde{G}(u) \tag{30}$$

where $\tilde{F}$ and $\tilde{G}$ are the maximal monotone operators on $\mathbb{R}^n \times \mathbb{R}^p$ defined as

$$\tilde{F}(u_1, u_2) = \begin{cases} F(u_1) \times \mathbb{R}^p & \text{if } u_2 = 0 \\ \emptyset & \text{otherwise,} \end{cases} \qquad \tilde{G}(u) = B^T G(Bu). \tag{31}$$

From these definitions, it is clear that $u = (u_1, u_2)$ solves (30) if and only if $u_2 = 0$ and $x = u_1$ solves (28). For an optimization problem (1) the reformulation amounts to solving

$$\text{minimize} \quad f(u_1) + \delta_{\{0\}}(u_2) + g(Au_1 + Cu_2),$$

where $\delta_{\{0\}}$ is the indicator function of $\{0\}$, and interpreting this as minimizing $\tilde{f}(u) + \tilde{g}(u)$ where

$$\tilde{f}(u) = f(u_1) + \delta_{\{0\}}(u_2), \qquad \tilde{g}(u) = g(Bu).$$

At first this simple reformulation seems pointless, as it is unrealistic to assume that a suitable matrix C is known. However we will see that, after simplifications, the matrix C is not needed to apply the algorithm.

We note that if F is μ-strongly monotone, then $\tilde{F}$ is μ-strongly monotone. If G is $(1/\beta)$-cocoercive, then $\tilde{G}$ is (γ^2/β)-cocoercive, and if G is β-Lipschitz continuous, then $\tilde{G}$ is (β/γ^2)-Lipschitz continuous.

To apply the DRS algorithm to (30) we need expressions for the resolvents of the two operators or their inverses. The resolvent of $\tau\tilde{F}$ is straightforward:

$$J_{\tau\tilde{F}}(u_1, u_2) = (J_{\tau F}(u_1), 0). \tag{32}$$

The resolvent of $\tau\tilde{G}$ follows from the first identity in (15):

$$J_{\tau\tilde{G}}(u) = \gamma^2 \left((\gamma^{-2}I - B^TB)u + B^T J_{(\tau/\gamma^2)G}(Bu)\right) = u + \gamma^2 B^T \left(J_{\sigma^{-1}G}(Bu) - Bu\right) \tag{33}$$

where we defined $\sigma = \gamma^2/\tau$. The second identity in (15) combined with (10) gives an expression for the resolvent of $(\tau\tilde{G})^{-1} = \tilde{G}^{-1}\tau^{-1}$:

$$J_{(\tau\tilde{G})^{-1}}(u) = \gamma^2 B^T J_{((\tau/\gamma^2)G)^{-1}}(Bu) = \gamma^2 B^T J_{G^{-1}\sigma}(Bu) = \tau B^T J_{\sigma G^{-1}}(\sigma Bu). \tag{34}$$

From the resolvent of $\tilde{G}^{-1}\tau^{-1}$ and the scaling rule (10) we obtain the resolvent of $\tau^{-1}\tilde{G}^{-1}$:

$$J_{\tau^{-1}\tilde{G}^{-1}}(u) = B^T J_{\sigma G^{-1}}(\gamma^2 Bu). \tag{35}$$

4.1 Relaxed DRS algorithm

The relaxed DRS iteration (24) applied to (30) involves the three steps

$$\begin{aligned}
\bar{u}^k &= J_{\tau\tilde{F}}(y^{k-1}) && \text{(36a)}\\
\bar{w}^k &= J_{(\tau\tilde{G})^{-1}}(2\bar{u}^k - y^{k-1}) && \text{(36b)}\\
y^k &= (1-\rho_k)y^{k-1} + \rho_k(\bar{u}^k - \bar{w}^k). && \text{(36c)}
\end{aligned}$$

The variables are vectors in $\mathbb{R}^n \times \mathbb{R}^p$, which we will partition as $\bar{u}^k = (\bar{u}_1^k, \bar{u}_2^k)$, $\bar{w}^k = (\bar{w}_1^k, \bar{w}_2^k)$, $y^k = (y_1^k, y_2^k)$. Substituting the expressions for the resolvents (32) and (34) gives

$$\begin{aligned}
\bar{u}^k &= (J_{\tau F}(y_1^{k-1}), 0)\\
\bar{w}^k &= \tau B^T J_{\sigma G^{-1}}(\sigma B(2\bar{u}^k - y^{k-1}))\\
y^k &= (1-\rho_k)y^{k-1} + \rho_k(\bar{u}^k - \bar{w}^k),
\end{aligned}$$

with $\sigma = \gamma^2/\tau$. Next we add two new variables $u^k = (1-\rho_k)u^{k-1} + \rho_k\bar{u}^k$ and $w^k = u^k - y^k$, and obtain

$$\begin{aligned}
\bar{u}^k &= (J_{\tau F}(y_1^{k-1}), 0) && \text{(38a)}\\
\bar{w}^k &= \tau B^T J_{\sigma G^{-1}}(\sigma B(2\bar{u}^k - y^{k-1})) && \text{(38b)}\\
u^k &= (1-\rho_k)u^{k-1} + \rho_k\bar{u}^k && \text{(38c)}\\
y^k &= (1-\rho_k)y^{k-1} + \rho_k(\bar{u}^k - \bar{w}^k) && \text{(38d)}\\
w^k &= u^k - y^k. && \text{(38e)}
\end{aligned}$$

The variable y^k can now be eliminated. If we subtract equation (38d) from equation (38c), we see that the w-update in equation (38e) can be written as

$$w^k = (1-\rho_k)(u^{k-1} - y^{k-1}) + \rho_k\bar{w}^k = (1-\rho_k)w^{k-1} + \rho_k\bar{w}^k.$$

Substituting $y^{k-1} = u^{k-1} - w^{k-1}$ in the first and second steps of the algorithm and removing (38d) then gives

$$\begin{aligned}
\bar{u}^k &= (J_{\tau F}(u_1^{k-1} - w_1^{k-1}), 0) && \text{(39a)}\\
\bar{w}^k &= \tau B^T J_{\sigma G^{-1}}(\sigma B w^{k-1} + \sigma B(2\bar{u}^k - u^{k-1})) && \text{(39b)}\\
u^k &= (1-\rho_k)u^{k-1} + \rho_k\bar{u}^k && \text{(39c)}\\
w^k &= (1-\rho_k)w^{k-1} + \rho_k\bar{w}^k. && \text{(39d)}
\end{aligned}$$

The expression on the first line shows that $\bar{u}_2^k = 0$ for all k. If we start with $u_2^0 = 0$, then line 3 shows that also $u_2^k = 0$ for all k. The expression on the second line shows that $\bar{w}^k$ is in the range of B^T for all k, *i.e.*, $\bar{w}^k = \tau B^T \bar{z}^k$ for some $\bar{z}^k$. If we choose w^0 in the range of B^T, then the last line shows that w^k is also in the range of B^T for all k, *i.e.*, $w^k = \tau B^T z^k$ for some z^k. Since $BB^T = \gamma^{-2} I$, the vectors $\bar{z}^k$ and z^k are uniquely defined and equal to $\bar{z}^k = (\gamma^2/\tau) B\bar{w}^k = \sigma B\bar{w}^k$ and $z^k = (\gamma^2/\tau)Bw^k = \sigma Bw^k$. If we make these substitutions and write $x^k = u_1^k$, $\bar{x}^k = \bar{u}_1^k$, then the iteration simplifies to

$$\bar{x}^k = J_{\tau F}(x^{k-1} - \tau A^T z^{k-1}) \tag{40a}$$

$$\bar{z}^k = J_{\sigma G^{-1}}(z^{k-1} + \sigma A(2\bar{x}^k - x^{k-1})) \tag{40b}$$

$$x^k = (1-\rho_k)x^{k-1} + \rho_k \bar{x}^k \tag{40c}$$

$$z^k = (1-\rho_k)z^{k-1} + \rho_k \bar{z}^k. \tag{40d}$$

The parameters τ and σ satisfy $\sigma\tau\|A\|^2 = \gamma^2\|A\|^2 \leq 1$.

To summarize, we have shown that the DRS method (36) for problem (30), started at an initial point $y^0 = (x^0 + \tau A^T z^0, \tau C^T z^0)$, is identical to the iteration (40), and that the iterates in the two algorithms are related as $y^k = (x^k + \tau A^T z^k, \tau C^T z^k)$. The correspondence between the other variables is that $\bar{u}^k = (\bar{x}^k, 0)$ and $\bar{w}^k = \tau B^T \bar{z}^k$. In particular, when written as (40), the DRS method does not require the matrix C. It is interesting to note that a combination of two facts allowed us to go from (39) to (40) and express the equations in terms of the first block A of B. In the product $B(2\bar{u}^k - u^{k-1})$ the matrix C is not needed because the second components of $\bar{u}^k$ and u^{k-1} are zero. The product Bw^{k-1} simplifies because w^{k-1} is in the range of B^T and $BB^T = \gamma^{-2}I$.

For $F = \partial f$ and $G = \partial g$, algorithm (40) is the PDHG algorithm (2). For general monotone operators it is the extension of the PDHG algorithm discussed in [3, 34].

4.2 Accelerated DRS algorithm

If F is μ-strongly monotone, the accelerated DRS algorithm (27) can be applied to (30):

$$u^k = J_{\tau_k \tilde{F}}(u^{k-1} - \tau_k w^{k-1})$$

$$w^k = J_{\tau_{k+1}^{-1}\tilde{G}^{-1}}(w^{k-1} + \frac{1}{\tau_{k+1}}(u^k + \theta_k(u^k - u^{k-1})))$$

with $\theta_k = 1/\sqrt{1 + 2\mu\tau_k}$ and $\tau_{k+1} = \theta_k\tau_k$ for $k = 1, 2, \ldots$. The variables u^k and w^k are in $\mathbb{R}^{n\times p}$ and will be partitioned as $u^k = (u_1^k, u_2^k)$, $w^k = (w_1^k, w_2^k)$. After substituting the expressions for the resolvents (32) and (34) we obtain

$$u^k = (J_{\tau_k F}(u_1^{k-1} - \tau_k w_1^{k-1}), 0)$$

$$w^k = B^T J_{\sigma_k G^{-1}}\left(\gamma^2 Bw^{k-1} + \sigma_k B(u^k + \theta_k(u^k - u^{k-1}))\right)$$

where $\sigma_k = \gamma^2/\tau_{k+1}$. From the first line, it is clear that $u_2^k = 0$ at all iterations. From the second line, w^k is in the range of B^T, so it can be written as $w^k = B^T z^k$ where $z_k = \gamma^2 Bw^k$. If we start at initial points that satisfy $u_2^0 = 0$ and $w^0 = B^T z^0$, and define $x^k = u_1^k$, the algorithm simplifies to

$$x^k = J_{\tau_k F}(x^{k-1} - \tau_k A^T z^{k-1}) \tag{41a}$$

$$z^k = J_{\sigma_k G^{-1}}\left(z^{k-1} + \sigma_k A(x^k + \theta_k(x^k - x^{k-1}))\right). \tag{41b}$$

This is the accelerated PDHG algorithm [26, Algorithm 2] applied to a monotone inclusion problem.

4.3 Convergence results

The reduction of PDHG to the DRS algorithm allows us to apply convergence results for DRS to PDHG. From the results in [7] discussed in §3 we obtain a convergence result for an inexact version of the relaxed

PDHG algorithm (40),

$$\begin{aligned}
\bar{x}^k &= J_{\tau F}(x^{k-1} - \tau A^T z^{k-1}) + d_k && \text{(42a)}\\
\bar{z}^k &= J_{\sigma G^{-1}}(z^{k-1} + \sigma A(2\bar{x}^k - x^{k-1})) + e_k && \text{(42b)}\\
x^k &= (1-\rho_k)x^{k-1} + \rho_k \bar{x}^k && \text{(42c)}\\
z^k &= (1-\rho_k)z^{k-1} + \rho_k \bar{z}^k. && \text{(42d)}
\end{aligned}$$

Suppose F and G are maximal monotone operators and (28) is solvable. If the conditions

$$\tau\sigma\|A\|^2 \le 1, \qquad \sum_{k=0}^{\infty} \rho_k(\|d_k\| + \|e_k\|) < \infty, \qquad \sum_{k=0}^{\infty} \rho_k(2-\rho_k) < \infty$$

are satisfied, then (x^k, z^k) and $(\bar{x}^k, \bar{z}^k)$ converge to a limit (x, z) that satisfies

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} F(x) \\ G^{-1}(z) \end{bmatrix}.$$

Applied to the PDHG algorithm (2) for solving (1), this gives Theorem 3.3 in [11]. The result from [11] was itself a significant improvement over the convergence results in earlier papers on PDHG, which mention convergence for $\tau\sigma\|A\|^2 < 1$ and $\rho_k = 1$; see [15, Theorem 2.4], [27, Theorem 2], [4, Theorem 1].

If F is strongly monotone and G is cocoercive, then the linear convergence result from [18] mentioned in §3.1 applies to the reformulated problem (30). If F is μ-strongly monotone and G is $(1/\beta)$-cocoercive, then $\tilde{F}$ is μ-strongly monotone and $\tilde{G}$ is (γ^2/β)-cocoercive. Therefore taking

$$\rho_k = \frac{2 + \gamma\sqrt{\mu/\beta}}{1 + \gamma\sqrt{\mu/\beta}}, \qquad \tau = \frac{\gamma}{\sqrt{\beta\mu}}, \qquad \sigma = \frac{\gamma^2}{\tau} = \gamma\sqrt{\beta\mu},$$

in the relaxed PDHG algorithm (40) gives a linear convergence rate with factor

$$\eta = \frac{1}{1 + \gamma\sqrt{\mu/\beta}}. \tag{43}$$

This can be compared with the convergence result for Algorithm 3 in [4], a version of the accelerated algorithm (41) with parameters

$$\theta_k = \frac{1}{1 + 2\gamma\sqrt{\mu/\beta}}, \qquad \tau_k = \frac{\gamma}{\sqrt{\beta\mu}}, \qquad \sigma_k = \gamma\sqrt{\beta\mu}$$

(in our notation). In [4, Theorem 3], the accelerated PDHG algorithm is shown to converge R-linearly with rate

$$\eta = \left(\frac{1}{1 + 2\gamma\sqrt{\mu/\beta}}\right)^{1/2}.$$

This is comparable in its dependence on $\sqrt{\mu/\beta}$ and higher than the rate (43), which holds for the relaxed non-accelerated algorithm.

Finally, if F is μ-strongly monotone, and no properties other than maximal monotonicity are assumed for G, then it follows from the result in [12] mentioned in §3.2 that the accelerated PDHG algorithm (41) converges with $\|x^k - x^\star\|^2 = O(1/k^2)$. This agrees with the result in [4, Theorem 2].

4.4 Related work

The connections of the PDHG algorithm with the proximal point algorithm, the DRS method, and the alternating direction method of multipliers (ADMM) have been the subject of several recent papers. He and Yuan [19] showed that algorithm (2) can be interpreted as a variant of the proximal point algorithm for finding a zero of a monotone operator [29] applied to the optimality condition (3). In the standard proximal point algorithm, the iterates x^{k+1} and z^{k+1} are defined as the solution x, z of

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) \\ \partial g^*(z) \end{bmatrix} + \frac{1}{\tau} \begin{bmatrix} x - x^k \\ z - z^k \end{bmatrix}.$$

In He and Yuan's modified algorithm, the last term on the right-hand side is replaced with

$$\begin{bmatrix} (1/\tau)I & -A^T \\ -A & (1/\sigma)I \end{bmatrix} \begin{bmatrix} x - x^k \\ z - z^k \end{bmatrix}, \tag{44}$$

where $\tau\sigma\|A\|_2^2 < 1$. This inequality ensures that the block matrix in (44) is positive definite, and the standard convergence theory of the proximal point algorithm applies. Shefi and Teboulle [31, §3.3] use a similar idea to interpret PDHG as an instance of a general class of algorithms derived from the proximal point method of multipliers [28]. Yan [35, Remark 1 in §3.2] gives an interpretation of the PDHG algorithm as the DRS algorithm applied to the primal-dual optimality conditions and using a quadratic norm defined by $I - \gamma^2 AA^T$.

In these papers the PDHG algorithm is interpreted as a modified proximal point or DRS method applied to problem (3), or a modified proximal method of multipliers applied to problem (1). This is different from the approach in the previous section, in which the algorithm is interpreted as the standard DRS algorithm applied to a reformulation of the original problem.

5 Primal form and application to non-convex problems

The starting point in §4 was the DRS algorithm in the "primal-dual" forms (24) and (27), *i.e.*, written in terms of the resolvents of F and G^{-1}. If instead we start from the original form of the DRS algorithm (23) or (26), we obtain a "primal" version of the PDHG algorithm expressed in terms of the resolvents of F and G. This variant was studied in [21, 22] for optimization problems of the form (1), in which g is allowed to be semiconvex (*i.e.*, has the property that $g(x) + (\lambda/2)\|x\|^2$ is convex for sufficiently large λ).

If G is maximal monotone, the resolvents of G^{-1} in the PDHG algorithm (41) can be written in terms of the resolvents of G using the identity (12). However in applications to nonconvex optimization, the identity (12), *i.e.*, the Moreau identity, does not hold. The direct derivation PDHG algorithm given in this section does not rely on the Moreau identity.

We work out the details for the accelerated DRS method (26), applied to (30):

$$\begin{aligned} u^k &= J_{\tau_k \tilde{F}}(y^{k-1}) \\ y^k &= J_{\tau_{k+1}\tilde{G}}((1+\theta_k)u^k - \theta_k y^{k-1}) + \theta_k(y^{k-1} - u^k). \end{aligned}$$

The formulas simplify if we introduce a variable $w^k = u^k - y^k$ and remove y^k:

$$\begin{aligned} u^k &= J_{\tau_k \tilde{F}}(u^{k-1} - w^{k-1}) \\ w^k &= u^k + \theta_k(u^k - u^{k-1}) + \theta_k w^{k-1} - J_{\tau_{k+1}\tilde{G}}(u^k + \theta_k(u^k - u^{k-1}) + \theta_k w^{k-1}). \end{aligned}$$

Substituting the expressions for the resolvents (32) and (33) then gives

$$\begin{aligned} u^k &= J_{\tau_k F}(u_1^{k-1} - w_1^{k-1}, 0) \\ w^k &= \gamma^2 B^T \left(B(u^k + \theta_k(u^k - u^{k-1}) + \theta_k w^{k-1}) - J_{\sigma_k^{-1} G}(B(u^k + \theta_k(u^k - u^{k-1}) + \theta_k w^{k-1})) \right) \end{aligned}$$

where $\sigma_k = \gamma^2/\tau_{k+1}$. We now apply the same argument as in the previous section. We assume $u_2^0 = 0$ and w^0 is in the range of B^T. Then $u_2^k = 0$ and w^k can be written as $w^k = \tau_{k+1} B^T z^k$ with $z^k = (\gamma^2/\tau_{k+1})Bw^k = \sigma_k B w^k$. After this change of variables, and with $x^k = u_1^k$, the algorithm reduces to

$$x^k = J_{\tau_k F}(x^{k-1} - \tau_k A^T z^{k-1}) \tag{45a}$$

$$z^k = z^{k-1} + \sigma_k A(x^k + \theta_k(x^k - x^{k-1})) - \sigma_k J_{\sigma_k^{-1} G}(\frac{1}{\sigma_k} z^{k-1} + A(x^k + \theta_k(x^k - x^{k-1}))). \tag{45b}$$

It is important to note that the steps in the derivation of algorithm 45 do not assume monotonicity of the operators F and G, since the expressions (32) and (33) hold in general. In particular, the expression for $J_{\tau\tilde{G}}$ was obtained from the first identity in (15), which, as we have seen, holds for general operators.

6 Three-operator extension of PDHG

Several extensions of the PDHG algorithm that minimize a sum of three functions $f(x) + g(Ax) + h(x)$, where h is differentiable, have recently been proposed [6,11,34,35]. A three-operator extension of the DRS algorithm was introduced by Davis and Yin in [12, Algorithm 1]. In this section, we use the technique of Section 4 to derive Yan's three-operator PDHG extension from the Davis-Yin algorithm.

Let F, G, and H be maximal monotone operators on $\mathbb{R}^n$ and assume that H is $1/\beta$-cocoercive. The Davis-Yin algorithm is a three-operator extension of DRS that solves the monotone inclusion problem

$$0 \in F(x) + G(x) + H(x)$$

via the iteration

$$\begin{aligned} x^k &= J_{tF}(y^{k-1}) \\ w^k &= J_{tG}(2x^k - y^{k-1} - tH(x^k)) \\ y^k &= y^{k-1} + \rho_k(w^k - x^k). \end{aligned}$$

Using the relation between J_{tG} and $J_{(tG)^{-1}}$ given in (12) and taking $\rho_k = 1$ for all k, we can write the algorithm as

$$\begin{aligned} x^k &= J_{tF}(y^{k-1}) \\ v^k &= J_{(tG)^{-1}}(2x^k - y^{k-1} - tH(x^k)) \\ y^k &= x^k - v^k - tH(x^k). \end{aligned}$$

Eliminating y^k, we obtain the iteration

$$x^k = J_{tF}(x^{k-1} - v^{k-1} - tH(x^{k-1})) \tag{46a}$$

$$v^k = J_{(tG)^{-1}}(2x^k - x^{k-1} + v^{k-1} + tH(x^{k-1}) - tH(x^k)). \tag{46b}$$

We will use the Davis-Yin algorithm to solve the monotone inclusion problem

$$0 \in F(x) + A^T G(Ax) + H(x), \tag{47}$$

where F and H are maximal monotone operators on $\mathbb{R}^n$, G is a maximal monotone operator on $\mathbb{R}^m$, A is a real $m \times n$ matrix, and H is $1/\beta$-cocoercive. Problem (47) is equivalent to the problem

$$0 \in \tilde{F}(u) + B^T \tilde{G}(Bu) + \tilde{H}(u), \tag{48}$$

where $\tilde{F}$ and $\tilde{G}$ are defined in equation (31), B is defined in equation (29), and $\tilde{H}$ is defined by

$$\tilde{H}(u_1, u_2) = H(u_1) \times \{0\}.$$

The Davis-Yin iteration (46) applied to problem (48) is

$$\begin{aligned} u^k &= J_{\tau\tilde{F}}(u^{k-1} - v^{k-1} - \tau\tilde{H}(u^{k-1})) \\ v^k &= J_{(\tau\tilde{G})^{-1}}(2u^k - u^{k-1} + v^{k-1} + \tau\tilde{H}(u^{k-1}) - \tau\tilde{H}(u^k)). \end{aligned}$$

The variables are vectors in $\mathbb{R}^n \times \mathbb{R}^p$, which we will partition as $u^k = (u_1^k, u_2^k)$, $v^k = (v_1^k, v_2^k)$. Substituting the expressions for the resolvents (32) and (34) gives

$$\begin{aligned} u^k &= (J_{\tau F}(u_1^{k-1} - v_1^{k-1} - \tau H(u_1^{k-1})), 0) \\ v^k &= \tau B^T J_{\sigma G^{-1}}(\sigma B(2u^k - u^{k-1} + v^{k-1} + \tau \tilde{H}(u^{k-1}) - \tau \tilde{H}(u^k))) \end{aligned}$$

with $\sigma = \gamma^2/\tau$. The expression on the first line shows that $u_2^k = 0$ for all $k \geq 1$. If we start with $u_2^0 = 0$, then $u_2^k = 0$ for all k. The expression on the second line shows that v^k is in the range of B^T for all $k \geq 1$. If we choose v^0 in the range of B^T, then v^k is in the range of B^T for all k, *i.e.*, $v^k = \tau B^T z^k$ for some z^k. Since $BB^T = \gamma^{-2} I$, the vectors z^k are uniquely defined and equal to $z^k = (\gamma^2/\tau) B v^k = \sigma B v^k$. If we make these substitutions and write $x^k = u_1^k$, then the iteration simplifies to

$$x^k = J_{\tau F}(x^{k-1} - \tau A^T z^{k-1} - \tau H(x^{k-1})) \tag{50a}$$

$$z^k = J_{\sigma G^{-1}}(z^{k-1} + \sigma A(2x^k - x^{k-1} + \tau H(x^{k-1}) - \tau H(x^k))). \tag{50b}$$

The parameters τ and σ satisfy $\sigma\tau\|A\|^2 = \gamma^2\|A\|^2 \leq 1$. In the case where $F = \partial f$, $G = \partial g$, and $H = \nabla h$, for some closed convex functions f and g and a differentiable convex function h, this is the PD3O algorithm introduced in [35, Equations 4(a)–4(c)].

The convergence results in [12] can now be applied directly to the PD3O algorithm. For example, it follows from Theorem 1.1 in [12] that if problem (47) has a solution and the step size restrictions $\tau < 2/\beta$ and $\sigma\tau\|A\|^2 \leq 1$ are satisfied then x^k converges to a solution of (47) as $k \to \infty$.

7 Separable structure

The parameters σ and τ in the PDHG algorithm must satisfy $\sigma\tau\|A\|^2 \leq 1$. When the problem has separable structure, it is possible to formulate an extension of the algorithm that uses more than two parameters. This can speed up the algorithm and make it easier to find suitable parameter values [26]. The extended algorithm can be derived by applying the standard PDHG algorithm after scaling the rows and columns of A and adjusting the expressions for the resolvents or proximal operators accordingly. It can also be derived directly from DRS as follows.

We consider a convex optimization problem with block-separable structure

$$\text{minimize} \quad \sum_{j=1}^n f_j(x_j) + \sum_{i=1}^m g_i(\sum_{j=1}^n A_{ij} x_j).$$

The functions $f_j : \mathbb{R}^{q_j} \to \mathbb{R}$ and $g_i : \mathbb{R}^{p_i} \to \mathbb{R}$ are closed and convex with nonempty domains, and A_{ij} is a matrix of size $p_i \times q_j$. To apply the DRS algorithm, we rewrite the problem as

$$\text{minimize} \quad \tilde{f}(u) + \tilde{g}(Bu) \tag{51}$$

where $\tilde{f}$ and $\tilde{g}$ are defined as

$$\tilde{f}(u_1, \ldots, u_n, u_{n+1}) = \sum_{j=1}^n f_j(\eta_j u_j) + \delta_{\{0\}}(u_{n+1}), \qquad \tilde{g}(v_1, \ldots, v_m) = \sum_{i=1}^m g_i(\gamma_i^{-1} v_i),$$

and B is an $m \times (n+1)$ block matrix

$$B = \begin{bmatrix} B_{11} & \cdots & B_{1n} & B_{1,n+1} \\ B_{21} & \cdots & B_{2n} & B_{2,n+1} \\ \vdots & & \vdots & \vdots \\ B_{m1} & \cdots & B_{mn} & B_{m,n+1} \end{bmatrix} = \begin{bmatrix} \gamma_1\eta_1 A_{11} & \gamma_1\eta_2 A_{12} & \cdots & \gamma_1\eta_n A_{1n} & C_1 \\ \gamma_2\eta_1 A_{21} & \gamma_2\eta_2 A_{22} & \cdots & \gamma_2\eta_n A_{2n} & C_2 \\ \vdots & \vdots & & \vdots & \vdots \\ \gamma_m\eta_1 A_{m1} & \gamma_m\eta_2 A_{m2} & \cdots & \gamma_m\eta_n A_{mn} & C_m \end{bmatrix}.$$

The positive coefficients γ_i, η_j and the matrices C_i are chosen so that $BB^T = I$. The matrices C_i will appear in the derivation of the algorithm but are ultimately not needed to execute it. It is sufficient to know that they exist, *i.e.*, that the first n block columns of B form a matrix with norm less than or equal to one.

The DRS algorithm with relaxation (24) for problem (51) is

$$\begin{aligned}\bar{u}^k &= \mathrm{prox}_{t\tilde{f}}(y^{k-1})\\ \bar{w}^k &= B^T\mathrm{prox}_{(t\tilde{g})^*}(B(2\bar{u}^k - y^{k-1}))\\ y^k &= (1-\rho_k)y^{k-1} + \rho_k(\bar{u}^k - \bar{w}^k).\end{aligned}$$

Here we applied the property that if $h(u) = \tilde{g}(Bu)$ and $BB^T = I$, then $\mathrm{prox}_{(th)^*} = B^T\mathrm{prox}_{(t\tilde{g})^*}(Bu)$ (see (14)). Next we add two variables u^k and w^k as follows:

$$\begin{aligned}\bar{u}^k &= \mathrm{prox}_{t\tilde{f}}(y^{k-1})\\ \bar{w}^k &= B^T\mathrm{prox}_{(t\tilde{g})^*}(B(2\bar{u}^k - y^{k-1}))\\ u^k &= (1-\rho_k)u^{k-1} + \rho_k\bar{u}^k\\ y^k &= (1-\rho_k)y^{k-1} + \rho_k(\bar{u}^k - \bar{w}^k)\\ w^k &= u^k - y^k.\end{aligned}$$

Eliminating y^k now results in

$$\begin{aligned}\bar{u}^k &= \mathrm{prox}_{t\tilde{f}}(u^{k-1} - w^{k-1})\\ \bar{w}^k &= B^T\mathrm{prox}_{(t\tilde{g})^*}(B(w^{k-1} + 2\bar{u}^k - u^{k-1}))\\ u^k &= (1-\rho_k)u^{k-1} + \rho_k\bar{u}^k\\ w^k &= (1-\rho_k)w^{k-1} + \rho_k\bar{w}^k.\end{aligned}$$

The proximal operators of $\tilde{f}$ and $\tilde{g}^*$ follow from the scaling property (11), the Moreau identity, and the separable structure of $\tilde{f}$ and $\tilde{g}$:

$$\begin{aligned}\mathrm{prox}_{t\tilde{f}}(u_1,\ldots,u_n,u_{n+1}) &= \big(\eta_1^{-1}\mathrm{prox}_{\tau_1 f_1}(\eta_1 u_1),\ \ldots,\ \eta_n^{-1}\mathrm{prox}_{\tau_n f_n}(\eta_n u_n), 0\big)\\ \mathrm{prox}_{(t\tilde{g})^*}(v_1,\ldots,v_m) &= \left(\frac{\gamma_1}{\sigma_1}\mathrm{prox}_{\sigma_1 g_1^*}(\frac{\sigma_1}{\gamma_1}v_1),\ \ldots,\ \frac{\gamma_m}{\sigma_m}\mathrm{prox}_{\sigma_m g_m^*}(\frac{\sigma_m}{\gamma_m}v_m)\right)\end{aligned}$$

where we define $\tau_j = t\eta_j^2$ and $\sigma_i = \gamma_i^2/t$. Substituting these formulas in the algorithm gives

$$\begin{aligned}\bar{u}_j^k &= \eta_j^{-1}\mathrm{prox}_{\tau_j f_j}(\eta_j(u_j^{k-1} - w_j^{k-1})), \quad j = 1,\ldots,n\\ \bar{u}_{n+1}^k &= 0\\ \bar{w}_j^k &= \sum_{i=1}^m \frac{\gamma_i}{\sigma_i}B_{ij}^T\,\mathrm{prox}_{\sigma_i g_i^*}\left(\frac{\sigma_i}{\gamma_i}\sum_{l=1}^{n+1}B_{il}(w_l^{k-1} + 2\bar{u}_l^k - u_l^{k-1})\right), \quad j = 1,\ldots,n+1\\ u^k &= (1-\rho_k)u^{k-1} + \rho_k\bar{u}^k\\ w^k &= (1-\rho_k)w^{k-1} + \rho_k\bar{w}^k.\end{aligned}$$

From these equations it is clear that if we start at an initial u^0 with last component $u_{n+1}^0 = 0$, then $u_{n+1}^k = \bar{u}_{n+1}^k = 0$ for all k. Furthermore, the third and last equation show that if w^0 is in the range of B^T, then w^k and $\bar{w}^k$ are in the range of B^T for all k, *i.e.*, they can be expressed as

$$w_j^k = \sum_{i=1}^m (t/\gamma_i)B_{ij}^T z_i^k, \qquad \bar{w}_j^k = \sum_{i=1}^m (t/\gamma_i)B_{ij}^T \bar{z}_i^k, \qquad j = 1,\ldots,n+1.$$

Since $BB^T = I$, the variables z_i^k and $\bar{z}_i^k$ are equal to

$$z_i^k = \frac{\gamma_i}{t}\sum_{j=1}^{n+1}B_{ij}w_j^k, \qquad \bar{z}_i^k = \frac{\gamma_i}{t}\sum_{j=1}^{n+1}B_{ij}\bar{w}_j^k, \qquad i = 1,\ldots,m.$$

If we make this change of variables and also define $\bar{x}_j^k = \eta_j \bar{u}_j^k$, $x_j^k = \eta_j u_j^k$, we obtain

$$\begin{aligned}
\bar{x}_j^k &= \mathrm{prox}_{\tau_j f_j}(x_j^{k-1} - \tau_j \sum_{i=1}^m A_{ij}^T z_i^{k-1}), \quad j = 1, \ldots, n \\
\bar{z}_i^k &= \mathrm{prox}_{\sigma_i g_i^*}(z_i^{k-1} + \sigma_i \sum_{l=1}^{n+1} A_{il}(2\bar{x}_l^k - x_l^{k-1})), \quad j = 1, \ldots, n+1 \\
x^k &= (1 - \rho_k)x^{k-1} + \rho_k \bar{x}^k \\
z^k &= (1 - \rho_k)z^{k-1} + \rho_k \bar{z}^k.
\end{aligned}$$

Since $\sigma_i \tau_j = \gamma_i^2 \eta_j^2$, the condition on the parameters σ_i and τ_j is that the matrix

$$\begin{bmatrix} \sqrt{\sigma_1\tau_1}A_{11} & \sqrt{\sigma_1\tau_2}A_{12} & \cdots & \sqrt{\sigma_1\tau_n}A_{1n} \\ \sqrt{\sigma_2\tau_1}A_{21} & \sqrt{\sigma_2\tau_2}A_{22} & \cdots & \sqrt{\sigma_2\tau_n}A_{2n} \\ \vdots & \vdots & & \vdots \\ \sqrt{\sigma_m\tau_1}A_{m1} & \sqrt{\sigma_m\tau_2}A_{m2} & \cdots & \sqrt{\sigma_m\tau_n}A_{mn} \end{bmatrix}$$

has norm less than or equal to one.

When $m = 1$, the condition on the stepsize parameters is $\tau_1 \sum_{j=1}^n \sigma_i A_{i1}^T A_{i1} \preceq I$. Condat [11, Algorithm 5.2] uses $\sigma_i = \sigma$ for all i and gives a convergence result for $\tau\sigma \sum_i A_{i1}^T A_{i1} \preceq I$ [11, Theorem 5.3]. Other authors use the weaker condition $\tau \sum_{i=1}^l \sigma_i \|A_{i1}\|^2 \leq 1$ [3, page 270], [34, page 678]. Pock and Chambolle in [26] discuss this method for separable functions f and g, and scalar A_{ij}.

8 Linearized ADMM

It was observed in [15, Figure 1] that linearized ADMM [24, 25] is equivalent to PDHG applied to the dual. (In [15] PDHG applied to the dual is called PDHGMp and linearized ADMM is called Split Inexact Uzawa.) This equivalence implies that linearized ADMM is also a special case of the Douglas–Rachford method. In this section we give a short proof of the equivalence between linearized ADMM and PDHG.

Linearized ADMM minimizes $f(x) + g(Ax)$, where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are closed convex functions and $A \in \mathbb{R}^{m \times n}$, via the following iteration:

$$\begin{aligned}
\tilde{x}^k &= \mathrm{prox}_{\tau f}(\tilde{x}^{k-1} - (\tau/\lambda)A^T(A\tilde{x}^{k-1} - z^{k-1} + u^{k-1})) \\
z^k &= \mathrm{prox}_{\lambda g}(A\tilde{x}^k + u^{k-1}) \\
u^k &= u^{k-1} + A\tilde{x}^k - z^k.
\end{aligned}$$

Here τ and λ satisfy $0 < \tau \leq \lambda/\|A\|^2$. For $k \geq 0$, define $x^k = \tilde{x}^{k+1}$, so that

$$z^k = \mathrm{prox}_{\lambda g}(Ax^{k-1} + u^{k-1}) \tag{52a}$$

$$u^k = u^{k-1} + Ax^{k-1} - z^k \tag{52b}$$

$$x^k = \mathrm{prox}_{\tau f}(x^{k-1} - (\tau/\lambda)A^T(Ax^{k-1} - z^k + u^k)). \tag{52c}$$

From equations (52b) and (52a), we have

$$u^k = u^{k-1} + Ax^{k-1} - \mathrm{prox}_{\lambda g}(Ax^{k-1} + u^{k-1}) = \lambda \mathrm{prox}_{\lambda^{-1}g^*}(\frac{1}{\lambda}(Ax^{k-1} + u^{k-1})). \tag{53}$$

Equation (52b) also implies that $2u^k - u^{k-1} = Ax^{k-1} - z^k + u^k$. Plugging this into equation (52c), we find that

$$x^k = \mathrm{prox}_{\tau f}(x^{k-1} - (\tau/\lambda)A^T(2u^k - u^{k-1})). \tag{54}$$

Let $y^k = (1/\lambda)u^k$ for all $k \geq 0$ and let $\sigma = 1/\lambda$. Then, from equations (53) and (54), we have

$$\begin{aligned}
y^k &= \mathrm{prox}_{\sigma g^*}(y^{k-1} + \sigma Ax^{k-1}) \\
x^k &= \mathrm{prox}_{\tau f}(x^{k-1} - \tau A^T(2y^k - y^{k-1})).
\end{aligned}$$

This is the same iteration that one obtains by using PDHG to solve the dual problem

$$\text{minimize} \quad f^*(-A^Tz) + g^*(z).$$

9 Conclusion

The main difficulty when applying the Douglas–Rachford splitting method to problem (1) is the presence of the matrix A, which can make the proximal operator of $g(Ax)$ expensive to compute, even when g itself has an inexpensive proximal operator. Evaluating the proximal operator $g(Ax)$ generally requires an iterative optimization algorithm. Several strategies are known to avoid this problem and implement the DRS method using only proximal operators of f and g, and the solution of linear equations. The first is to reformulate the problem as

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array}$$

and apply the DRS method to $h(x, y) + \delta_V(x, y)$ where $h(x, y) = f(x) + g(y)$ and δ_V is the indicator function of the subspace $V = \{(x, y) \mid Ax = y\}$. The proximal operator of δ_V is the projection on V, and can be evaluated by solving a linear equation with coefficient $AA^T + I$. This is known as Spingarn's method [14, 32, 33].

A second option is to reformulate the problem as

$$\begin{array}{ll} \text{minimize} & f(u) + g(y) \\ \text{subject to} & \begin{bmatrix} I \\ A \end{bmatrix} x = \begin{bmatrix} u \\ y \end{bmatrix} \end{array}$$

and solve it via the alternating direction method of multipliers (ADMM), which is equivalent to DRS applied to its dual [14, 16]. Each iteration of ADMM applied to this problem requires an evaluation of the proximal operators of f and g, and the solution of a linear equation with coefficient $AA^T + I$.

A third option is to apply the DRS method for operators to the primal-dual optimality conditions (3). The resolvent of the second term on the right-hand side of (3) requires the proximal operators of f and g. The resolvent of the linear term can be computed by solving a linear equation with coefficient $t^{-2}I + AA^T$. This method and other primal-dual applications of the DRS splitting are discussed in [23].

In these three different applications of the DRS method, the cost of solving the set of linear equations usually determines the overall complexity. The PDHG method has the important advantage that it only requires multiplications with A and its transpose, but not the solution of linear equations. Although it is often seen as a generalization of the DRS method, the derivation in this paper shows that it can in fact be interpreted as the DRS method applied to an equivalent reformulation of the problem. This observation gives new insight in the PDHG algorithm, allows us to apply existing convergence theory for the DRS method, and greatly simplifies the formulation and analysis of extensions.

References

1. H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* Springer, 2011.
2. H. H. Bauschke, P. L. Combettes, and D. R. Luke. Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. *Journal of the Optical Society of America A*, 19:1334–1345, 2002.
3. R. I. Boţ, E. R. Csetnek, A. Heinrich, and C. Hendrich. On the convergence rate improvement of a primal-dual splitting algorithm for solving monotone inclusion problems. *Mathematical Programming*, 150:251–279, 2015.
4. A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.
5. A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, pages 161–319, 2016.
6. A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal-dual algorithm. *Mathematical Programming, Series A*, 159:253–287, 2016.
7. P. L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5–6):475–504, 2004.
8. P. L. Combettes and J.-C. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):564–574, 2007.
9. P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.
10. P. L. Combettes and I. Yamada. Compositions and convex combinations of averaged nonexpansive operators. *Journal of Mathematical Analysis and Applications*, 425:55–70, 2015.

11. L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.
12. D. Davis and W. Yin. A three-operator splitting scheme and its optimization applications, 2015. `arxiv.org/abs/1504.01032`.
13. D. Davis and W. Yin. Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions. *Mathematics of Operations Research*, 42:783–805, 2017.
14. J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.
15. E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.
16. D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, Studies in Mathematics and Its Applications, pages 299–331. North-Holland, 1983.
17. P. Giselsson. Tight global linear convergence rate bounds for Douglas-Rachford splitting. *Journal of Fixed Point Theory and Applications*, 19:2241–2270, 2017.
18. P. Giselsson and S. Boyd. Linear convergence and metric selection for Douglas-Rachford splitting and ADMM. *IEEE Transactions on Automatic Control*, 62(2):532–544, 2017.
19. B. He and X. Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.
20. P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.
21. T. Mölenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers. Low rank priors for color image regularization. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 126–140. Springer, 2015.
22. T. Mölenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers. The primal-dual hybrid gradient method for semiconvex splittings. *SIAM Journal on Imaging Sciences*, 8(2):827–857, 2015.
23. D. O'Connor and L. Vandenberghe. Primal-dual decomposition by operator splitting and applications to image deblurring. *SIAM Journal on Imaging Sciences*, 7(3):1724–1754, 2014.
24. H. Ouyang, N. He, and A. Gray. Stochastic ADMM for nonsmooth optimization. *arXiv preprint arXiv:1211.0632*, 2012.
25. N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
26. T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings 2011 IEEE International Conference on Computer Vision*, pages 1762–1769, 2011.
27. T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV)*, pages 1133–1140, 2009.
28. R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.
29. R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control and Opt.*, 14(5):877–898, August 1976.
30. R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 1998.
31. R. Shefi and M. Teboulle. Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM Journal on Optimization*, 24(1):269–297, 2014.
32. J. E. Spingarn. Partial inverse of a monotone operator. *Applied Mathematics and Optimization*, 10:247–265, 1983.
33. J. E. Spingarn. Applications of the method of partial inverses to convex programming: decomposition. *Mathematical Programming*, 32:199–223, 1985.
34. B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38:667–681, 2013.
35. M. Yan. A new primal-dual algorithm for minimizing the sum of three functions with a linear operator. *Journal of Scientific Computing*, 2018.