Bregman three-operator splitting methods

Xin Jiang Lieven Vandenberghe

January 26, 2022

Abstract

1 Introduction

We discuss proximal splitting methods for optimization problems in the form

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

where f, g, and h are convex functions, h is differentiable, and $f + h$ and g are closed functions with nonempty domains. This general problem covers a wide variety of applications in machine learning, signal and image processing, operations research, control, and many other fields [CP11b, PB13, KP15, CP16a]. In this paper, we consider proximal splitting methods based on Bregman distances for solving (1) and some interesting special cases of (1).

Recently, several primal–dual first-order methods have been proposed for the three-term problem (1): the Condat–Vũ algorithm [Con13, Vũ13], PD3O [Yan18], and PDDY [SCMR20]. Algorithms for some special cases of (1) are also of interest. These include the Chambolle–Pock algorithm, also known as the primal–dual hybrid gradient (PDHG) method [CP11a, CP16b] (when $h = 0$), the Loris–Verhoeven algorithm [LV11, CHZ13, DST15] (when $f = 0$), the proximal gradient algorithm (when $g = 0$), and the Davis–Yin splitting algorithm [DY15] (when $A = I$). All these methods handle the nonsmooth functions f and g via the standard Euclidean proximal operator.

To further improve the efficiency of proximal algorithms, generalized proximal operators based on generalized Bregman distances have been proposed and incorporated in many methods [CT93, Eck93, Gül94, AT06, Tse08, BBT17, BSTV18, Teb18]. Bregman distances offer two potential benefits. First, the Bregman distance can help build a more accurate local optimization model around the current iterate. This is often interpreted as a form of preconditioning. For example, diagonal or quadratic preconditioning [PC11, JLLO19, LXY21] has been shown to improve the convergence rate of PDHG, as well as the accuracy of the computed solution [ADH$^+$21]. As a second potential benefit, a Bregman proximal operator of a function may be easier to compute than the standard Euclidean proximal operator, and therefore reduce the complexity per iteration of an optimization algorithm. Recent applications of this kind include optimal transport problems [CLMW19], optimization over nonnegative trigonometric polynomials [CV18], and sparse semidefinite programming [JV21].

Extending standard proximal methods and their convergence analysis to Bregman distances is not straightforward because some fundamental properties of the Euclidean proximal operator no longer hold for Bregman proximal operators. An example is the Moreau decomposition which relates the (Euclidean) proximal operators of a closed convex function and its conjugate [Mor65]. Another example is the simple relation between the proximal operators of a function g and the composition with a linear function $g(Ax)$ when AA^T is a multiple of the identity. This composition rule was exploited in [OV18] to establish the equivalence between some well-known first-order proximal methods for problem (1) with $A = I$ and with general A. The main purpose of this paper is to present new Bregman extensions and convergence results for the Condat–Vũ and PD3O algorithms.

Contributions We first discuss two Bregman primal–dual first-order methods for (1), namely, the *Bregman primal Condat–Vũ algorithm* and the *Bregman dual Condat–Vũ algorithm*. These algorithms correspond to two variants of the Condat–Vũ algorithm [Con13, Vũ13], extended to use Bregman proximal operators in the primal and dual spaces. The Bregman primal Condat–Vũ algorithm first appeared in [CP16b, Algorithm 1]; see also [YHA21]. We give a new derivation of this method and its dual variant, by applying the Bregman proximal point method to the primal–dual optimality conditions. Based on the interpretation, we provide a unified framework for the convergence analysis of the two variants, and show an $O(1/k)$ ergodic convergence rate, consistent with previous results for Euclidean proximal operators in [Con13, Vũ13] and Bregman proximal operators in [CP16b]. We also give a convergence result for the primal and dual iterates. We further introduce an easily implemented line search technique for selecting better stepsizes in the Bregman dual Condat–Vũ algorithm for problems with equality constraints. The proposed backtracking procedure is similar to the technique in [MP18] for the special setting of PDHG with Euclidean proximal operators, but with important differences even in this special case. Again, we give a detailed analysis of the algorithm with line search and recover the $O(1/k)$ ergodic rate of convergence for related algorithms in [MP18, JV21]. Last, we propose a Bregman extension for PD3O and establish an ergodic convergence result.

Outline The rest of the paper is organized as follows. In Section 2 we review some well-known first-order proximal methods, and build connections between them via the two steps of completion [OV18] and reduction (*i.e.*, setting some parts of problem (1) to zero). Section 3 provides some necessary background on Bregman distances. In Section 4 we discuss the Bregman primal Condat–Vũ algorithm and the Bregman dual Condat–Vũ algorithm, and analyze their convergence. The line search technique and its convergence are discussed in Section 5. In Section 6 we extend PD3O to a Bregman proximal method, and also show its convergence. Section 7 contains results of a numerical experiment.

Notation The standard inner product of vectors x and y is denoted by $\langle x, y\rangle = x^T y$. The Euclidean norm of a vector x is denoted by $\|x\| = \langle x, x\rangle^{1/2}$. Other norms will be distinguished by a subscript.

The *proximal operator* or *proximal mapping* of a closed convex function $f : \mathbf{R}^p \to \mathbf{R}$ is defined as

$$\operatorname{prox}_f(y) = \underset{x}{\operatorname{argmin}} \left(f(x) + \frac{1}{2}\|x - y\|^2 \right). \tag{2}$$

If f is closed and convex, the minimizer in the definition exists and is unique for all y [Mor65]. We will call (2) the *standard* or the *Euclidean proximal operator* when we need to distinguish it from Bregman proximal operators defined in Section 3.

We discuss primal–dual methods that solve (1) and the dual problem

$$\text{maximize} \quad -(f+h)^*(-A^T z) - g^*(z), \tag{3}$$

where $(f+h)^*$ and g^* are the conjugates of $f+h$ and g:

$$(f+h)^*(y) = \sup_x \left(\langle y, x\rangle - f(x) - h(x)\right), \qquad g^*(z) = \sup_x \left(\langle z, x\rangle - g(x)\right).$$

We will assume that the primal–dual optimality conditions

$$0 \in \partial f(x) + \nabla h(x) + A^T z, \qquad 0 \in \partial g^*(z) - Ax$$

are solvable. Here ∂f and ∂g^* are the subdifferentials of f and g^*. We often write the optimality conditions as

$$0 \in \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} \partial f(x) + \nabla h(x) \\ \partial g^*(z) \end{bmatrix}. \tag{4}$$

The Lagrangian of (1) will be defined as $\mathcal{L}(x,z) = f(x) + h(x) + \langle z, Ax\rangle - g^*(z)$. The objective functions in (1) and the dual problem (3) can be expressed as

$$\sup_z \mathcal{L}(x,z) = f(x) + h(x) + g(Ax), \qquad \inf_x \mathcal{L}(x,z) = -(f+h)^*(-A^T z) - g^*(z).$$

Solutions $\hat{x}$, $\hat{z}$ of the optimality conditions (4) form a saddle-point of $\mathcal{L}$, *i.e.*, satisfy

$$\sup_z \mathcal{L}(\hat{x}, z) = \mathcal{L}(\hat{x}, \hat{z}) = \inf_x \mathcal{L}(x, \hat{z}), \tag{5}$$

2 First-order proximal algorithms: survey and connections

In this section, we discuss several first-order proximal algorithms and their connections. We start with four three-operator splitting algorithms for problem (1): the primal and dual variants of the Condat–Vũ algorithm [Con13, Vũ13], the primal–dual three-operator (PD3O) algorithm [Yan18], and the primal–dual Davis–Yin (PDDY) algorithm [SCMR20]. For each of the four algorithms, we make connections with other classical first-order proximal algorithms, using reduction (*i.e.*, setting some parts in (1) to zero) and the "completion" reformulation [OV18]. The main results are summarized in Figures 1–4.

2.1 Condat–Vũ three-operator splitting algorithm

We start with the (primal) Condat–Vũ three-operator splitting algorithm, which was proposed independently by Condat [Con13] and Vũ [Vũ13],

$$x^{(k+1)} = \operatorname{prox}_{\tau f}\big(x^{(k)} - \tau(A^T z^{(k)} + \nabla h(x^{(k)}))\big) \tag{6a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}\big(z^{(k)} + \sigma A(2x^{(k+1)} - x^{(k)})\big). \tag{6b}$$

The stepsizes σ and τ must satisfy

$$\sigma\tau\|A\|_2^2 + \tau L \leq 1, \tag{7}$$

where $\|A\|_2$ is the spectral norm of A, and L is the Lipschitz constant of ∇h with respect to the Euclidean norm. Many other first-order proximal algorithms can be viewed as special cases

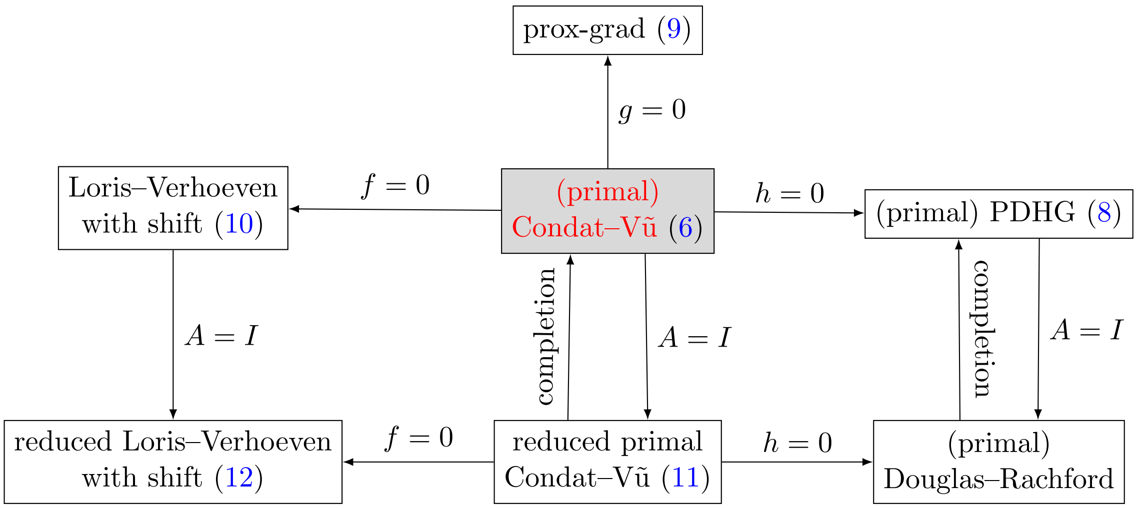

Figure 1: Summary of first-order proximal methods reduced from primal Condat–Vũ algorithm.

of (6), and their connections are summarized in Figure 1. When $h = 0$, algorithm (6) reduces to the so-called (primal) primal–dual hybrid gradient (PDHG) method [PCBC09, CP11a, CP16b], or PDHGMu in [EZC10]:

$$x^{(k+1)} = \operatorname{prox}_{\tau f}\big(x^{(k)} - \tau A^T z^{(k)})\big) \tag{8a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}\big(z^{(k)} + \sigma A(2x^{(k+1)} - x^{(k)})\big). \tag{8b}$$

When $g = 0$ in (6) (and assuming $z^{(0)} = 0$), we obtain the proximal gradient algorithm:

$$x^{(k+1)} = \operatorname{prox}_{\tau f}(x^{(k)} - \tau \nabla h(x^{(k)})). \tag{9}$$

When $f = 0$, we obtain a variant of the Loris–Verhoeven algorithm [LV11, CHZ13, DST15],

$$x^{(k+1)} = x^{(k)} - \tau(A^T z^{(k)} + \nabla h(x^{(k)})) \tag{10a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}\big((I - \sigma\tau AA^T)z^{(k)} + \sigma A(x^{(k+1)} - \tau\nabla h(x^{(k)})\big). \tag{10b}$$

We refer to this as *Loris–Verhoeven with shift*, for reasons that will be clarified later. Furthermore, when $A = I$ in PDHG, we obtain the classical Douglas–Rachford splitting (DRS) algorithm [LM79, EB92, CP07]. Conversely, the "completion" technique in [OV18] shows that PDHG coincides with DRS applied to a reformulation of the problem. Similarly, when $A = I$ in the primal Condat–Vũ algorithm (6), we obtain a new algorithm

$$x^{(k+1)} = \operatorname{prox}_{\tau f}\big(x^{(k)} - \tau(z^{(k)} + \nabla h(x^{(k)}))\big) \tag{11a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}\big(z^{(k)} + \sigma(2x^{(k+1)} - x^{(k)})\big). \tag{11b}$$

To the best of our knowledge, this algorithm has never been proposed in an independent manner, and we will refer to it as the *reduced primal Condat–Vũ algorithm*. Conversely, algorithm (11) reverts to (6) via the "completion" trick. We can also set $f = 0$ in (11) or $A = I$ in (10), and obtain the *reduced Loris–Verhoeven algorithm with shift*:

$$x^{(k+1)} = x^{(k)} - \tau(z^{(k)} + \nabla h(x^{(k)})) \tag{12a}$$

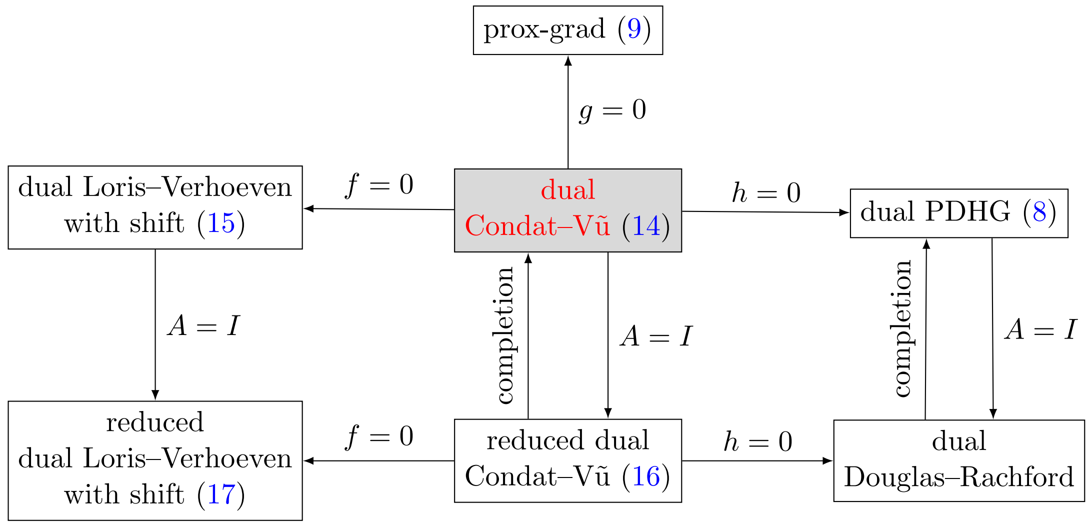

Figure 2: Summary of first-order proximal methods reduced from dual Condat–Vũ algorithm.

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}\big((1-\sigma\tau)z^{(k)} + \sigma(x^{(k+1)} - \tau\nabla h(x^{(k)}))\big). \tag{12b}$$

Recall the stepsize condition $\tau(\sigma + L) \le 1$, which implies $\tau \le 1/L$ and $\sigma\tau < 1$. Only for the sake of comparison, we set $\sigma = 1/\tau$ in (12), and obtain the iterations:

$$x^{(k+1)} = \operatorname{prox}_{\tau g}\big(x^{(k)} - \tau\nabla h(x^{(k-1)})\big) - \tau\nabla h(x^{(k)}) + \tau\nabla h(x^{(k-1)}). \tag{13}$$

This update appears to be the proximal gradient method with shift: The gradient term in (13) is evaluated at the previous iterate, not the most up-to-date one, and then it is compensated by an additional correction term $\tau(\nabla h(x^{(k-1)}) - \nabla h(x^{(k)}))$. This small difference turns out be critical in the requirement for the stepsizes; see discussion at the end of Section 6. Finally, due to the absence of f in (12), it is not clear how to apply the "completion" trick to (12) to obtain (10).

Condat [Con13] also discusses a variant of (6), which we will call the dual Condat–Vũ algorithm:

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}(z^{(k)} + \sigma A x^{(k)}) \tag{14a}$$
$$x^{(k+1)} = \operatorname{prox}_{\tau f}\big(x^{(k)} - \tau(A^T(2z^{(k+1)} - z^{(k)}) + \nabla h(x^{(k)}))\big). \tag{14b}$$

Figure 2 summarizes the proximal algorithms reduced from (14). When $h = 0$, algorithm (14) reduces to PDHG applied to the dual of (1) (with $h = 0$), which is shown to be equivalent to linearized ADMM [OHG12, PB13] (also called Split Inexact Uzawa in [EZC10]). Setting $g = 0$ in (14) yields the proximal gradient algorithm (9). When $f = 0$, we obtain a new algorithm:

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}(z^{(k)} + \sigma A x^{(k)}) \tag{15a}$$
$$x^{(k+1)} = x^{(k)} - \tau(A^T(2z^{(k+1)} - z^{(k)}) + \nabla h(x^{(k)})). \tag{15b}$$

Following the previous naming convention, we call it *dual Loris–Verhoeven algorithm with shift*, but we shall notice that (15) is not algorithm (10) applied to the dual problem. Furthermore, setting $A = I$ in (14) gives the *reduced dual Condat–Vũ algorithm*:

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}(z^{(k)} + \sigma x^{(k)}) \tag{16a}$$

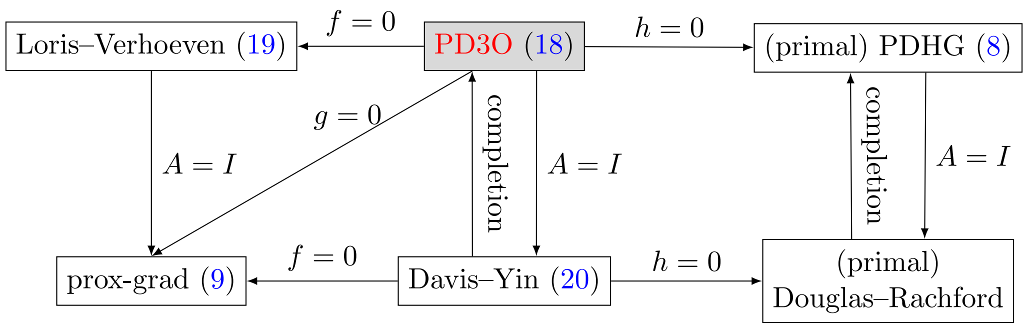

Figure 3: Summary of first-order proximal algorithms reduced from PD3O.

$$x^{(k+1)} = \mathrm{prox}_{\tau f}\big(x^{(k)} - \tau(2z^{(k+1)} - z^{(k)} + \nabla h(x^{(k)}))\big). \tag{16b}$$

Conversely, applying the "completion" trick to (16) recovers (14). Similarly, setting $A = I$ in dual PDHG gives dual DRS, *i.e.*, DRS with f and g switched, and conversely, the "completion" trick recovers dual PDHG from dual DRS. We can also set $A = I$ in (15) or $f = 0$ in (16), and obtain the *reduced dual Loris–Verhoeven algorithm with shift*:

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}(z^{(k)} + \sigma x^{(k)}) \tag{17a}$$
$$x^{(k+1)} = x^{(k)} - \tau(2z^{(k+1)} - z^{(k)} + \nabla h(x^{(k)})). \tag{17b}$$

Note that in (16) and (17), the stepsize condition (7) requires $\tau(\sigma + L) \le 1$, which implies $\tau \le 1/L$ and $\sigma\tau < 1$.

2.2 Primal–dual three-operator (PD3O) splitting algorithm

The third diagram, Figure 3, starts with the primal–dual three-operator (PD3O) splitting algorithm [Yan18]

$$x^{(k+1)} = \mathrm{prox}_{\tau f}(x^{(k)} - \tau(A^T z^{(k)} + \nabla h(x^{(k)}))) \tag{18a}$$
$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}(z^{(k)} + \sigma A(2x^{(k+1)} - x^{(k)} + \tau\nabla h(x^{(k)}) - \tau\nabla h(x^{(k+1)}))). \tag{18b}$$

Compared with the Condat–Vũ algorithm (6), PD3O seems to have slightly more complicated updates and larger complexity per iteration, but the requirement for the stepsizes is looser: $\sigma\tau\|A\|_2^2 \le 1$ and $\tau \le 1/L$. When $h = 0$, (18) reduces to the (primal) PDHG (8). The classical proximal gradient algorithm (9) can be obtained by setting $g = 0$. When $f = 0$, it reduces to the iterations

$$x^{(k+1)} = x^{(k)} - \tau(A^T z^{(k)} + \nabla h(x^{(k)})) \tag{19a}$$
$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}\big((I - \sigma\tau AA^T)z^{(k)} + \sigma A(x^{(k+1)} - \tau\nabla h(x^{(k+1)}))\big). \tag{19b}$$

This algorithm was discovered independently as the Loris–Verhoeven algorithm [LV11], the primal–dual fixed point algorithm based on proximity operator (PDFP^{2}O) [CHZ13], and the proximal alternating predictor corrector (PAPC) [DST15]. Comparison with (10) reveals a minor difference between these two algorithms: the gradient term in the z-update is taken at the newest primal iterate $x^{(k+1)}$ in Loris–Verhoeven (19) and at the previous point $x^{(k)}$ in the shifted version. This difference is inherited in the proximal gradient algorithm and its shifted version (12).

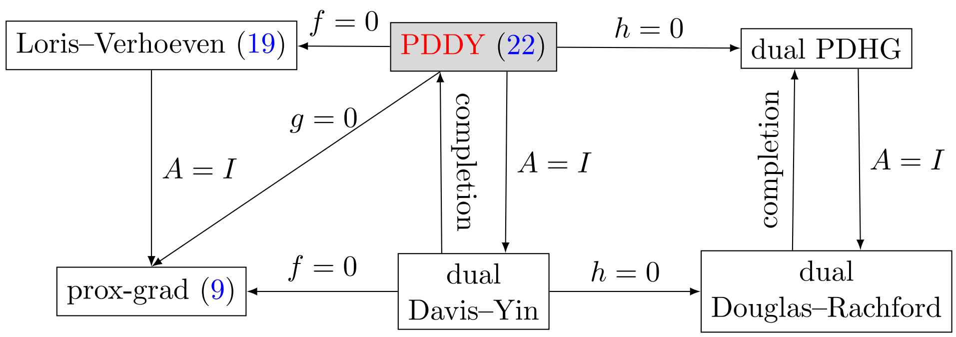

Figure 4: Summary of first-order proximal algorithms reduced from PDDY.

Furthermore, when $A = I$ and $\sigma = 1/\tau$ in PD3O, we recover the well-known Davis–Yin splitting (DYS) algorithm [DY15]

$$x^{(k+1)} = \mathrm{prox}_{\tau f}(x^{(k)} - \tau z^{(k)} - t\nabla h(x^{(k)})) \tag{20a}$$

$$z^{(k+1)} = \mathrm{prox}_{\tau^{-1} g^*}\Big(z^{(k)} + \frac{1}{\tau}(2x^{(k+1)} - x^{(k)}) + \nabla h(x^{(k)}) - \nabla h(x^{(k+1)})\Big). \tag{20b}$$

Note again that the only difference between (20) and the reduced Condat–Vũ (11) algorithm is the additional correction term $\nabla h(x^{(k)}) - \nabla h(x^{(k+1)})$. We can also set $A = I$ in (19) and obtain the iterations

$$x^{(k+1)} = x^{(k)} - \tau(z^{(k)} + \nabla h(x^{(k)})) \tag{21a}$$

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}\big((1 - \sigma\tau)z^{(k)} + \sigma(x^{(k+1)} - \tau\nabla h(x^{(k+1)})\big). \tag{21b}$$

The stepsize conditions require $\sigma\tau \leq 1$ and $\tau \leq 1/L$. Thus we can set $\sigma = 1/\tau$ and apply Moreau decomposition. The resulting algorithm is exactly the proximal gradient method. The only difference in the z-update between (12) and (21) is the position at which the gradient of h is taken. It makes more sense to use the most up-to-date iterate $x^{(k+1)}$ when evaluating the gradient of h, and this choice also allows a larger stepsize τ.

2.3 Primal–dual Davis–Yin (PDDY) splitting algorithm

The core algorithm in Figure 4 is the primal–dual Davis–Yin (PDDY) splitting algorithm [SCMR20]

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}(z^{(k)} + \sigma A x^{(k)}) \tag{22a}$$

$$x^{(k+1)} = \mathrm{prox}_{\tau f}\big(x^{(k)} - \tau A^T(2z^{(k+1)} - z^{(k)}) - \tau\nabla h(x^{(k)} + \tau A^T(z^{(k)} - z^{(k+1)}))\big). \tag{22b}$$

The requirement for stepsizes is the same as that in PD3O: $\sigma\tau\|A\|_2^2 \leq 1$ and $\tau \leq 1/L$. Figure 4 is almost identical to Figure 3 with the roles of f and g exchanged. When $h = 0$, PDDY reduces to the dual PDHG. In addition, when $A = I$ and $\sigma = 1/\tau$, PDDY reduces to the Davis–Yin algorithm, but with f and g exchanged. Similarly, when $h = 0$, $A = I$ and $\sigma = 1/\tau$, PDDY reverts to the Douglas–Rachford algorithm with f and g switched.

We have seen that the middle and right parts of Figure 4 is those of Figure 3 with f and g switched. However, when one of the functions f or g disappears, the algorithms reduced from PD3O and PDDY are exactly the same. In particular, when $f = 0$, PDDY reduces to the Loris–Verhoeven algorithm.

3 Bregman distances

In this section we give the definition of Bregman proximal operators and the basic properties that will be used in the paper. We refer the interested reader to [CZ97] for an in-depth discussion of Bregman distances, their history, and applications.

Let ϕ be a convex function with a domain that has nonempty interior, and assume ϕ is continuously differentiable on $\mathbf{int}(\mathbf{dom}\,\phi)$. The *generalized distance* (or *Bregman distance*) generated by the *kernel function* ϕ is defined as the function

$$d(x,y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x-y \rangle,$$

with domain $\mathbf{dom}\,d = \mathbf{dom}\,\phi \times \mathbf{int}(\mathbf{dom}\,\phi)$. The corresponding *Bregman proximal operator* of a function f is

$$\begin{aligned} \mathrm{prox}_f^{\phi}(y,a) &= \operatorname*{argmin}_x \left(f(x) + \langle a, x\rangle + d(x,y)\right) && (23) \\ &= \operatorname*{argmin}_x \left(f(x) + \langle a, x\rangle + \phi(x) - \langle \nabla\phi(y), x\rangle\right). && (24) \end{aligned}$$

It is assumed that for every a and every $y \in \mathbf{int}(\mathbf{dom}\,\phi)$ the minimizer $\hat{x} = \mathrm{prox}_f^{\phi}(y,a)$ is unique and in $\mathbf{int}(\mathbf{dom}\,\phi)$.

The distance generated by the kernel $\phi(x) = (1/2)\|x\|^2$ is the squared Euclidean distance $d(x,y) = (1/2)\|x-y\|^2$. The corresponding Bregman proximal operator is the standard proximal operator applied to $y - a$:

$$\mathrm{prox}_f^{\phi}(y,a) = \mathrm{prox}_f(y-a).$$

For this distance, closedness and convexity of f guarantee that the proximal operator is well defined. The questions of existence and uniqueness are more complicated for general Bregman distances. There are no simple general conditions that guarantee that for every a and every $y \in \mathbf{int}(\mathbf{dom}\,\phi)$ the generalized proximal operator (23) is uniquely defined and in $\mathbf{int}(\mathbf{dom}\,\phi)$. Some sufficient conditions are provided (see, for example, [Bub15, Section 4.1], [BBT17, Assumption A]), but they may be quite restrictive or difficult to verify in practice. In applications, however, the Bregman proximal operator is used with specific combinations of f and ϕ, for which the minimization problem in (23) is particularly easy to solve. In those applications, existence and uniqueness of the solution follow directly from the closed-form solution or availability of a fast algorithm to compute it. A typical example will be provided in Section 7; see (81).

From the expression (24) we see that $\hat{x} = \mathrm{prox}_f^{\phi}(y,a)$ satisfies

$$\nabla\phi(y) - \nabla\phi(\hat{x}) - a \in \partial f(\hat{x}). \tag{25}$$

Equivalently, by definition of subgradient,

$$\begin{aligned} f(x) + \langle a, x\rangle &\geq f(\hat{x}) + \langle a, \hat{x}\rangle + \langle \nabla\phi(y) - \nabla\phi(\hat{x}), x - \hat{x}\rangle \\ &= f(\hat{x}) + \langle a, \hat{x}\rangle + d(\hat{x},y) + d(x,\hat{x}) - d(x,y) && (26) \end{aligned}$$

for all $x \in \mathbf{dom}\,f \cap \mathbf{dom}\,\phi$.

4 Bregman Condat–Vũ three-operator splitting algorithms

We now discuss two Bregman three-operator splitting algorithms for the problem (1). The algorithms use a generalized distance d_{p} in the primal space, generated by a kernel ϕ_{p}, and a generalized distance d_{d} in the dual space, generated by a kernel ϕ_{d}. The first algorithm is

$$x^{(k+1)} = \mathrm{prox}_{\tau f}^{\phi_{\mathrm{p}}}\big(x^{(k)}, \tau A^T z^{(k)} + \tau \nabla h(x^{(k)})\big) \tag{27a}$$

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}}\big(z^{(k)}, -\sigma A(2x^{(k+1)} - x^{(k)})\big) \tag{27b}$$

and will be referred to as the *Bregman primal Condat–Vũ algorithm*. The second algorithm will be called the *Bregman dual Condat–Vũ algorithm*:

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}}(z^{(k)}, -\sigma A x^{(k)}) \tag{28a}$$

$$x^{(k+1)} = \mathrm{prox}_{\tau f}^{\phi_{\mathrm{p}}}(x^{(k)}, \tau A^T(2z^{(k+1)} - z^{(k)}) + \tau \nabla h(x^{(k)})). \tag{28b}$$

The two algorithms need starting points $x^{(0)} \in \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{p}}) \cap \mathbf{dom}\,h$, and $z^{(0)} \in \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{d}})$. Conditions on stepsizes σ, τ will be specified later. When Euclidean distances are used for the primal and dual proximal operators, the two algorithms reduce to the primal and dual variants of the Condat–Vũ algorithm (6) and (14), respectively. Algorithm (27) has been proposed in [CP16a]. Here we discuss it together with (27) here under a unified framework.

In Section 4.1 we show that the proposed algorithms can be interpreted as the Bregman proximal point method applied to a monotone inclusion problem. In Section 4.2 we analyze their convergence. In Section 4.3 we discuss the connections between the two algorithms and other Bregman proximal splitting methods.

Assumptions Throughout Section 4 we make the following assumptions. The kernel functions ϕ_{p} and ϕ_{d} are 1-strongly convex with respect to norms $\|\cdot\|_{\mathrm{p}}$ and $\|\cdot\|_{\mathrm{d}}$, respectively:

$$d_{\mathrm{p}}(x, x') \geq \frac{1}{2}\|x - x'\|_{\mathrm{p}}^2, \qquad d_{\mathrm{d}}(z, z') \geq \frac{1}{2}\|z - z'\|_{\mathrm{d}}^2 \tag{29}$$

for all $(x, x') \in \mathbf{dom}\,d_{\mathrm{p}}$ and $(z, z') \in \mathbf{dom}\,d_{\mathrm{d}}$. The assumption that the strong convexity constants are equal to one can be made without loss of generality, by scaling the norms (or distances) if needed. We also assume that the function $L\phi_{\mathrm{p}} - h$ is convex for some $L > 0$. More precisely, $\mathbf{dom}\,\phi_{\mathrm{p}} \subseteq \mathbf{dom}\,h$ and

$$h(x) - h(x') - \langle \nabla h(x'), x - x' \rangle \leq L d_{\mathrm{p}}(x, x') \quad \text{for all } (x, x') \in \mathbf{dom}\,d_{\mathrm{p}}. \tag{30}$$

Note that this assumption is looser than the one in [CP16a, Equation (4)]. We denote by $\|A\|$ the matrix norm

$$\|A\| = \sup_{u \neq 0, v \neq 0} \frac{\langle v, Au \rangle}{\|v\|_{\mathrm{d}} \|u\|_{\mathrm{p}}} = \sup_{u \neq 0} \frac{\|Au\|_{\mathrm{d},*}}{\|u\|_{\mathrm{p}}} = \sup_{v \neq 0} \frac{\|A^T v\|_{\mathrm{p},*}}{\|v\|_{\mathrm{d}}}, \tag{31}$$

where $\|\cdot\|_{\mathrm{p},*}$ and $\|\cdot\|_{\mathrm{d},*}$ are the dual norms of $\|\cdot\|_{\mathrm{p}}$ and $\|\cdot\|_{\mathrm{d}}$.

It is also assumed that the primal–dual optimality conditions (4) have a solution $(x^\star, z^\star)$ with $x^\star \in \mathbf{dom}\,\phi_{\mathrm{p}}$ and $z^\star \in \mathbf{dom}\,\phi_{\mathrm{d}}$. This point $(x^\star, z^\star)$ is a saddle-point of the Lagrangian $\mathcal{L}$.

4.1 Derivation from Bregman proximal point method

The Bregman Condat–Vũ algorithms (27) and (28) can be viewed as applications of the Bregman proximal point algorithm to the optimality conditions (4). This interpretation extends the derivation of the Bregman PDHG algorithm from the Bregman proximal point algorithm given in [JV21]. Similar ideas were first used by He and Yuan in their interpretation of PDHG (8) as a "preconditioned" proximal point algorithm [HY12].

The Bregman proximal point algorithm [Eck93, CZ97, Gül94] is an algorithm for monotone inclusion problems $0 \in F(u)$. The update $u^{(k+1)}$ in one iteration of the algorithm is defined as the solution of the inclusion

$$\nabla\phi(u^{(k)}) - \nabla\phi(u^{(k+1)}) \in F(u^{(k+1)}),$$

where ϕ is a Bregman kernel function. Applied to (4), with a kernel function ϕ_{pd}, the algorithm generates a sequence $(x^{(k)}, z^{(k)})$ defined by

$$\nabla\phi_{\mathrm{pd}}(x^{(k)}, z^{(k)}) - \nabla\phi_{\mathrm{pd}}(x^{(k+1)}, z^{(k+1)}) \in \begin{bmatrix} A^T z^{(k+1)} + \partial f(x^{(k+1)}) + \nabla h(x^{(k+1)}) \\ -Ax^{(k+1)} + \partial g^*(z^{(k+1)}) \end{bmatrix}. \tag{32}$$

4.1.1 Primal–dual Bregman distances

We introduce four possible primal–dual kernel functions: the functions

$$\phi_+(x,z) = \frac{1}{\tau}\phi_{\mathrm{p}}(x) + \frac{1}{\sigma}\phi_{\mathrm{d}}(z) + \langle z, Ax\rangle, \qquad \phi_-(x,z) = \frac{1}{\tau}\phi_{\mathrm{p}}(x) + \frac{1}{\sigma}\phi_{\mathrm{d}}(z) - \langle z, Ax\rangle,$$

where $\sigma, \tau > 0$, and the functions

$$\phi_{\mathrm{dcv}}(x,z) = \phi_+(x,z) - h(x), \qquad \phi_{\mathrm{pcv}}(x,z) = \phi_-(x,z) - h(x).$$

The subscripts in ϕ_+ and ϕ_- refer to the sign in front of the inner product term $\langle z, Ax\rangle$. The subscripts in ϕ_{pcv} and ϕ_{dcv} indicate the algorithm (Bregman primal or dual Condat-Vũ) for which these distances will be relevant. If these kernel functions are convex, they generate the following Bregman distances. The distances generated by ϕ_+ and ϕ_- are

$$d_+(x,z;x',z') = \frac{1}{\tau}d_{\mathrm{p}}(x,x') + \frac{1}{\sigma}d_{\mathrm{d}}(z,z') + \langle z - z', A(x - x')\rangle \tag{33}$$

$$d_-(x,z;x',z') = \frac{1}{\tau}d_{\mathrm{p}}(x,x') + \frac{1}{\sigma}d_{\mathrm{d}}(z,z') - \langle z - z', A(x - x')\rangle, \tag{34}$$

respectively, and the distances generated by ϕ_{dcv} and ϕ_{pcv} are

$$d_{\mathrm{dcv}}(x,z;x',z') = d_+(x,z;x',z') - h(x) + h(x') + \langle \nabla h(x'), x - x'\rangle \tag{35}$$

$$d_{\mathrm{pcv}}(x,z;x',z') = d_-(x,z;x',z') - h(x) + h(x') + \langle \nabla h(x'), x - x'\rangle. \tag{36}$$

We now show that ϕ_+ and ϕ_- are convex if

$$\sigma\tau\|A\|^2 \le 1 \tag{37}$$

and strongly convex if $\sigma\tau\|A\|^2 < 1$, and that the functions ϕ_{dcv} and ϕ_{pcv} are convex if

$$\sigma\tau\|A\|^2 + \tau L \le 1 \tag{38}$$

and strongly convex if $\sigma\tau\|A\|^2 + \tau L < 1$.

Proof. To show that the kernel functions ϕ_+ and ϕ_- are convex, we show that d_+ and d_- are nonnegative. Suppose $\sigma\tau\|A\|^2 \le \delta^2$ with $0 < \delta \le 1$. Then (29) and the arithmetic–geometric mean inequality imply that

$$\begin{aligned}
\left|\langle z - z', A(x - x')\rangle\right| &\le \|A\| \|z - z'\|_{\mathrm{d}} \|x - x'\|_{\mathrm{p}} \\
&\le \frac{\delta}{\sqrt{\sigma\tau}} \|z - z'\|_{\mathrm{d}} \|x - x'\|_{\mathrm{p}} \\
&\le \frac{\delta}{2\tau} \|x - x'\|_{\mathrm{p}}^2 + \frac{\delta}{2\sigma} \|z - z'\|_{\mathrm{p}}^2 \\
&\le \frac{\delta}{\tau} d_{\mathrm{p}}(x, x') + \frac{\delta}{\sigma} d_{\mathrm{d}}(z, z'). \qquad (39)
\end{aligned}$$

Therefore

$$\begin{aligned}
d_\pm(x, z; x', z') &= \frac{1}{\tau} d_{\mathrm{p}}(x, x') + \frac{1}{\sigma} d_{\mathrm{d}}(z, z') \pm \langle z - z', A(x - x')\rangle \\
&\ge \frac{1 - \delta}{\tau} d_{\mathrm{p}}(x, x') + \frac{1 - \delta}{\sigma} d_{\mathrm{d}}(z, z') \\
&\ge \frac{1 - \delta}{2\tau} \|x - x'\|_{\mathrm{p}}^2 + \frac{1 - \delta}{2\sigma} \|z - z'\|_{\mathrm{d}}^2.
\end{aligned}$$

With $\delta = 1$, this shows convexity of ϕ_+ and ϕ_-; with $\delta < 1$, strong convexity.

Similarly, if $\sigma\tau\|A\|^2 \le \delta(\delta - \tau L)$, with $0 < \delta \le 1$, then

$$\begin{aligned}
\left|\langle z - z', A(x - x')\rangle\right| &\le \frac{\sqrt{\delta(\delta - \tau L)}}{\sqrt{\sigma\tau}} \|z - z'\|_{\mathrm{d}} \|x - x'\|_{\mathrm{p}} \\
&\le \frac{\delta - \tau L}{2\tau} \|x - x'\|_{\mathrm{p}}^2 + \frac{\delta}{2\sigma} \|z - z'\|_{\mathrm{d}}^2 \\
&\le \frac{\delta - \tau L}{\tau} d_{\mathrm{p}}(x, x') + \frac{\delta}{\sigma} d_{\mathrm{d}}(z, z')
\end{aligned}$$

and

$$\begin{aligned}
d_{\mathrm{dcv/pcv}}(x, z; x', z') &= \frac{1}{\tau} d_{\mathrm{p}}(x, x') + \frac{1}{\sigma} d_{\mathrm{d}}(z, z') \pm \langle z - z', A(x - x')\rangle \\
&\quad - h(x) + h(x') + \langle \nabla h(x'), x - x'\rangle \\
&\ge (\frac{1 - \delta}{\tau} + L) d_{\mathrm{p}}(x, x') + \frac{1 - \delta}{\sigma} d_{\mathrm{d}}(z, z') - h(x) + h(x') + \langle \nabla h(x'), x - x'\rangle \\
&\ge \frac{1 - \delta}{\tau} d_{\mathrm{p}}(x, x') + \frac{1 - \delta}{\sigma} d_{\mathrm{d}}(z, z').
\end{aligned}$$

□

4.1.2 Bregman Condat-Vũ algorithms from proximal point method

The Bregman primal Condat–Vũ algorithm is the Bregman proximal point method with the kernel function $\phi_{\mathrm{pd}} = \phi_{\mathrm{pcv}}$. If we take $\phi_{\mathrm{pd}} = \phi_{\mathrm{pcv}}$ in (32), we obtain two coupled inclusions determine $x^{(k+1)}$, $z^{(k+1)}$. The first one is

$$0 \in \frac{1}{\tau}(\nabla\phi_{\mathrm{p}}(x^{(k+1)}) - \nabla\phi_{\mathrm{p}}(x^{(k)})) - A^T(z^{(k+1)} - z^{(k)}) - \nabla h(x^{(k+1)}) + \nabla h(x^{(k)})$$

$$
\begin{aligned}
&\quad + A^T z^{(k+1)} + \partial f(x^{(k+1)}) + \nabla h(x^{(k+1)}) \\
&= \frac{1}{\tau}(\nabla\phi_{\mathrm{p}}(x^{(k+1)}) - \nabla\phi_{\mathrm{p}}(x^{(k)})) + A^T z^{(k)} + \nabla h(x^{(k)}) + \partial f(x^{(k+1)}).
\end{aligned}
$$

This shows that $x^{(k+1)}$ solves the optimization problem

$$
\text{minimize} \quad f(x) + \langle A^T z^{(k)} + \nabla h(x^{(k)}), x\rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)}).
$$

The solution is the x-update (27a) in the Bregman primal Condat–Vũ method. The second inclusion is

$$
\begin{aligned}
0 &\in \frac{1}{\sigma}(\nabla\phi_{\mathrm{d}}(z^{(k+1)}) - \nabla\phi_{\mathrm{d}}(z^{(k)})) - A(x^{(k+1)} - x^{(k)}) - Ax^{(k+1)} + \partial g^*(z^{(k+1)}) \\
&= \frac{1}{\sigma}(\nabla\phi_{\mathrm{d}}(z^{(k+1)}) - \nabla\phi_{\mathrm{d}}(z^{(k)})) - A(2x^{(k+1)} - x^{(k)}) + \partial g^*(z^{(k+1)}).
\end{aligned}
$$

This shows that $z^{(k+1)}$ solves the optimization problem

$$
\text{minimize} \quad g^*(z) - \langle z, A(2x^{(k+1)} - x^{(k)})\rangle + \frac{1}{\sigma} d_{\mathrm{d}}(z, z^{(k)}).
$$

The solution is the z-update (27b).

Choosing $\phi_{\mathrm{pd}} = \phi_{\mathrm{dcv}}$ in (32) yields the Bregman dual Condat–Vũ algorithm (28). Substituting $\phi_{\mathrm{pd}} = \phi_{\mathrm{dcv}}$ in (32) gives the inclusions

$$
\begin{aligned}
0 &\in \frac{1}{\tau}(\nabla\phi_{\mathrm{p}}(x^{(k+1)}) - \nabla\phi_{\mathrm{p}}(x^{(k)})) + A^T(z^{(k+1)} - z^{(k)}) - \nabla h(x^{(k+1)}) + \nabla h(x^{(k)}) \\
&\quad + A^T z^{(k+1)} + \partial f(x^{(k+1)}) + \nabla h(x^{(k+1)}) \\
&= \frac{1}{\tau}(\nabla\phi_{\mathrm{p}}(x^{(k+1)}) - \nabla\phi_{\mathrm{p}}(x^{(k)})) + A^T(2z^{(k+1)} - z^{(k)}) + \nabla h(x^{(k)}) + \partial f(x^{(k+1)})
\end{aligned}
$$

and

$$
\begin{aligned}
0 &\in \frac{1}{\sigma}(\nabla\phi_{\mathrm{d}}(z^{(k+1)}) - \nabla\phi_{\mathrm{d}}(z^{(k)})) + A(x^{(k+1)} - x^{(k)}) - Ax^{(k+1)} + \partial g^*(z^{(k+1)}) \\
&= \frac{1}{\sigma}(\nabla\phi_{\mathrm{d}}(z^{(k+1)}) - \nabla\phi_{\mathrm{d}}(z^{(k)})) - Ax^{(k)} + \partial g^*(z^{(k+1)}).
\end{aligned}
$$

The second inclusion shows that $z^{(k+1)}$ is given by the z-update (28a). Given $z^{(k+1)}$, one can solve the first inclusion for $x^{(k+1)}$ and obtains the x-update (28b).

4.2 Convergence analysis

The derivation in Section 4.1 allows us to apply existing convergence theory for the Bregman proximal point method for monotone inclusions to the proposed algorithms (27) and (28). The literature on the Bregman proximal point method for monotone inclusions [Eck93, Gül94, CZ97] focuses on the convergence of iterates, and this generally requires additional assumptions on ϕ_{p} and ϕ_{d} (beyond the assumptions of convexity made in Section 4.1). In this section we present a self-contained convergence analysis for the duality gap associated with (4), and give a direct proof of an $O(1/k)$ rate of ergodic convergence. We also give a self-contained proof of convergence of the iterates $x^{(k)}$, $z^{(k)}$.

We make the assumptions listed in Section 4.1: the strong convexity assumption (29) for the primal and dual kernels ϕ_{p} and ϕ_{d}, and the *relative smoothness* property (30) of the function h. We assume that the stepsizes σ, τ satisfy (38), and that the primal–dual optimality condition (5) has a solution $(x^\star, z^\star) \in \mathbf{dom}\, \phi_{\mathrm{p}} \times \mathbf{dom}\, \phi_{\mathrm{d}}$.

For the sake of brevity we combine the analysis of the Bregman primal and the Bregman dual Condat-Vũ algorithms. In the following, d, $\tilde{d}$, $\tilde{\phi}$ are defined as

$$d = d_-, \qquad \tilde{d} = d_{\mathrm{pcv}}, \qquad \tilde{\phi} = \phi_{\mathrm{pcv}}$$

for the Bregman primal Condat–Vũ algorithm (27) and

$$d = d_+, \qquad \tilde{d} = d_{\mathrm{dcv}}, \qquad \tilde{\phi} = \phi_{\mathrm{dcv}}$$

for the Bregman dual Condat–Vũ algorithm (28).

4.2.1 One-iteration analysis

We first show that the iterates $x^{(k+1)}$, $z^{(k+1)}$ generated by the Bregman Condat–Vũ algorithms (27) and (28) satisfy

$$\begin{aligned} &\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\ &\quad \leq \; d(x, z; x^{(k)}, z^{(k)}) - d(x, z; x^{(k+1)}, z^{(k+1)}) - \tilde{d}(x^{(k+1)}, z^{(k+1)}; x^{(k)}, z^{(k)}) \end{aligned} \tag{40}$$

for all $x \in \mathbf{dom}\, f \cap \mathbf{dom}\, \phi_{\mathrm{p}}$ and $z \in \mathbf{dom}\, g^* \cap \mathbf{dom}\, \phi_{\mathrm{d}}$.

Proof. We write (27) and (28) in a unified notation as

$$x^{(k+1)} = \mathrm{prox}_{\tau f}^{\phi_{\mathrm{p}}}(x^{(k)}, \tau(A^T\tilde{z} + \nabla h(x^{(k)}))) \tag{41a}$$

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}}(z^{(k)}, -\sigma A\tilde{x}) \tag{41b}$$

where $\tilde{x}$ and $\tilde{z}$ are defined in the following table:

Bregman primal Condat–Vũ algorithm	$\tilde{x} = 2x^{(k+1)} - x^{(k)}$	$\tilde{z} = z^{(k)}$
Bregman dual Condat–Vũ algorithm	$\tilde{x} = x^{(k)}$	$\tilde{z} = 2z^{(k+1)} - z^{(k)}$.

The optimality condition (26) for the proximal operator evaluation (41a) is that

$$\tau(f(x^{(k+1)}) - f(x)) \leq d_{\mathrm{p}}(x, x^{(k)}) - d_{\mathrm{p}}(x^{(k+1)}, x^{(k)}) - d_{\mathrm{p}}(x, x^{(k+1)}) + \tau\langle A^T\tilde{z} + \nabla h(x^{(k)}), x - x^{(k+1)}\rangle$$

for all $x \in \mathbf{dom}\, f \cap \mathbf{dom}\, \phi_{\mathrm{p}}$. The optimality condition for (41b) is that

$$\sigma(g^*(z^{(k+1)}) - g^*(z)) \leq d_{\mathrm{d}}(z, z^{(k)}) - d_{\mathrm{d}}(z^{(k+1)}, z^{(k)}) - d_{\mathrm{d}}(z, z^{(k+1)}) - \sigma\langle z - z^{(k+1)}, A\tilde{x}\rangle$$

for all $z \in \mathbf{dom}\, g^* \cap \mathbf{dom}\, \phi_{\mathrm{d}}$. Combining the two inequalities gives

$$\begin{aligned} &\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\ &\quad = \; f(x^{(k+1)}) - f(x) + h(x^{(k+1)}) - h(x) + g^*(z^{(k+1)}) - g^*(z) + \langle A^T z, x^{(k+1)}\rangle - \langle z^{(k+1)}, Ax\rangle \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{\tau}\Big(d_{\rm p}(x,x^{(k)}) - d_{\rm p}(x,x^{(k+1)}) - d_{\rm p}(x^{(k+1)},x^{(k)})\Big) \\
&\quad + \frac{1}{\sigma}\Big(d_{\rm d}(z,z^{(k)}) - d_{\rm d}(z,z^{(k+1)}) - d_{\rm d}(z^{(k+1)},z^{(k)})\Big) \\
&\quad + h(x^{(k+1)}) - h(x) + \langle \nabla h(x^{(k)}), x - x^{(k+1)}\rangle \\
&\quad + \langle A^T\tilde z, x - x^{(k+1)}\rangle - \langle z - z^{(k+1)}, A\tilde x\rangle + \langle A^T z, x^{(k+1)}\rangle - \langle z^{(k+1)}, Ax\rangle \\
&\leq \frac{1}{\tau}\Big(d_{\rm p}(x,x^{(k)}) - d_{\rm p}(x,x^{(k+1)}) - d_{\rm p}(x^{(k+1)},x^{(k)})\Big) \\
&\quad + \frac{1}{\sigma}\Big(d_{\rm d}(z,z^{(k)}) - d_{\rm d}(z,z^{(k+1)}) - d_{\rm d}(z^{(k+1)},z^{(k)})\Big) \\
&\quad + h(x^{(k+1)}) - h(x^{(k)}) + \langle \nabla h(x^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\
&\quad + \langle A^T\tilde z, x - x^{(k+1)}\rangle - \langle z - z^{(k+1)}, A\tilde x\rangle + \langle A^T z, x^{(k+1)}\rangle - \langle z^{(k+1)}, Ax\rangle. \qquad (42)
\end{aligned}
$$

The second inequality follows from convexity of h. Substituting the expressions for $\tilde x$ and $\tilde z$ in the Bregman primal Condat–Vũ algorithm (27), we obtain on the last line of (42)

$$
\begin{aligned}
&\langle A^T\tilde z, x - x^{(k+1)}\rangle - \langle z - z^{(k+1)}, A\tilde x\rangle + \langle A^T z, x^{(k+1)}\rangle - \langle z^{(k+1)}, Ax\rangle \\
&\quad = \langle z^{(k)}, A(x - x^{(k+1)})\rangle - \langle z - z^{(k+1)}, A(2x^{(k+1)} - x^{(k)})\rangle + \langle A^T z, x^{(k+1)}\rangle - \langle z^{(k+1)}, Ax\rangle \\
&\quad = \langle z^{(k)} - z^{(k+1)}, A(x - x^{(k+1)})\rangle + \langle z - z^{(k+1)}, A(x^{(k)} - x^{(k+1)})\rangle \\
&\quad = -\langle z - z^{(k)}, A(x - x^{(k)})\rangle + \langle z - z^{(k+1)}, A(x - x^{(k+1)})\rangle + \langle z^{(k+1)} - z^{(k)}, A(x^{(k+1)} - x^{(k)})\rangle.
\end{aligned}
$$

If we substitute the expressions for $\tilde x$ and $\tilde z$ in the Bregman dual algorithm, the last line of (42) becomes

$$
\begin{aligned}
&\langle A^T\tilde z, x - x^{(k+1)}\rangle - \langle z - z^{(k+1)}, A\tilde x\rangle + \langle A^T z, x^{(k+1)}\rangle - \langle z^{(k+1)}, Ax\rangle \\
&\quad = \langle A^T(z - z^{(k)}), x - x^{(k)}\rangle - \langle A^T(z - z^{(k+1)}), x - x^{(k+1)}\rangle - \langle A^T(z^{(k+1)} - z^{(k)}), x^{(k+1)} - x^{(k)}\rangle.
\end{aligned}
$$

Therefore, for both algorithms, the duality gap (42) can be bounded as

$$
\begin{aligned}
&\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\
&\quad \leq \frac{1}{\tau}d_{\rm p}(x,x^{(k)}) + \frac{1}{\sigma}d_{\rm d}(z,z^{(k)}) \mp \langle z - z^{(k)}, A(x - x^{(k)})\rangle \\
&\qquad - \Big(\frac{1}{\tau}d_{\rm p}(x,x^{(k+1)}) + \frac{1}{\sigma}d_{\rm d}(z,z^{(k+1)}) \mp \langle z - z^{(k+1)}, A(x - x^{(k+1)})\rangle\Big) \\
&\qquad - \Big(\frac{1}{\tau}d_{\rm p}(x^{(k+1)},x^{(k)}) + \frac{1}{\sigma}d_{\rm d}(z^{(k+1)},z^{(k)}) \mp \langle z^{(k+1)} - z^{(k)}, A(x^{(k+1)} - x^{(k)})\rangle\Big) \\
&\qquad + h(x^{(k+1)}) - h(x^{(k)}) - \langle \nabla h(x^{(k)}), x^{(k+1)} - x^{(k)}\rangle,
\end{aligned}
$$

if we select the minus sign in $\mp$ for the Bregman primal Condat–Vũ algorithm, and the plus sign for the Bregman dual Condat–Vũ algorithm. For the primal method,

$$
\begin{aligned}
&\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\
&\quad \leq d_-(x,z;x^{(k)},z^{(k)}) - d_-(x,z;x^{(k+1)},z^{(k+1)}) - d_{\rm pcv}(x^{(k+1)},z^{(k+1)};x^{(k)},z^{(k)}).
\end{aligned}
$$

For the dual method,

$$
\begin{aligned}
&\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\
&\quad \leq d_+(x,z;x^{(k)},z^{(k)}) - d_+(x,z;x^{(k+1)},z^{(k+1)}) - d_{\rm dcv}(x^{(k+1)},z^{(k+1)};x^{(k)},z^{(k)}).
\end{aligned}
$$

□

4.2.2 Ergodic convergence

We define averaged iterates

$$x_{\rm avg}^{(k)} = \frac{1}{k}\sum_{i=1}^{k} x^{(i)}, \qquad z_{\rm avg}^{(k)} = \frac{1}{k}\sum_{i=1}^{k} z^{(i)} \tag{43}$$

for $k \geq 1$. Note that $x_{\rm avg}^{(k)} \in \mathbf{dom}\, f \cap \mathbf{int}(\mathbf{dom}\,\phi_{\rm p})$ and $z_{\rm avg}^{(k)} \in \mathbf{dom}\, g^* \cap \mathbf{int}(\mathbf{dom}\,\phi_{\rm d})$. We show that

$$\mathcal{L}(x_{\rm avg}^{(k)}, z^\star) - \mathcal{L}(x^\star, z_{\rm avg}^{(k)}) \leq \frac{2}{k}\Big(\frac{1}{\tau} d_{\rm p}(x^\star, x^{(0)}) + \frac{1}{\sigma} d_{\rm d}(z^\star, z^{(0)})\Big) \quad \text{for } k \geq 1, \tag{44}$$

Proof. From (40), since $\mathcal{L}(u, v)$ is convex in u and concave in v,

$$\begin{aligned}
\mathcal{L}(x_{\rm avg}^{(k)}, z) - \mathcal{L}(x, z_{\rm avg}^{(k)}) &\leq \frac{1}{k}\sum_{i=1}^{k}\big(\mathcal{L}(x^{(i)}, z) - \mathcal{L}(x, z^{(i)})\big) \\
&\leq \frac{1}{k}\big(d(x, z; x^{(0)}, z^{(0)}) - d(x, z; x^{(k)}, z^{(k)})\big) \\
&\leq \frac{1}{k} d(x, z; x^{(0)}, z^{(0)}) \\
&\leq \frac{2}{k}\Big(\frac{1}{\tau} d_{\rm p}(x, x^{(0)}) + \frac{1}{\sigma} d_{\rm d}(z, z^{(0)})\Big)
\end{aligned}$$

for all $x \in \mathbf{dom}\, f \cap \mathbf{int}\,\mathbf{dom}\,\phi_{\rm p}$ and $z \in \mathbf{dom}\, g^* \cap \mathbf{int}\,\mathbf{dom}\,\phi_{\rm d}$. The last step follows from (39). □

4.2.3 Monotonicity properties

For $x = x^\star$, $z = z^\star$, the left-hand side of (40) is nonnegative and therefore

$$d(x^\star, z^\star; x^{(k+1)}, z^{(k+1)}) \leq d(x^\star, z^\star; x^{(k)}, z^{(k)}) - \tilde{d}(x^{(k+1)}, z^{(k+1)}; x^{(k)}, z^{(k)}) \tag{45}$$

for $k \geq 0$. Hence $d(x^\star, z^\star; x^{(k+1)}, z^{(k+1)}) \leq d(x^\star, z^\star; x^{(k)}, z^{(k)})$ and

$$d(x^\star, z^\star; x^{(k)}, z^{(k)}) \leq d(x^\star, z^\star; x^{(0)}, z^{(0)}). \tag{46}$$

The inequality (45) also implies that

$$\sum_{i=0}^{k} \tilde{d}(x^{(i+1)}, z^{(i+1)}; x^{(i)}, z^{(i)}) \leq d(x^\star, z^\star; x^{(0)}, z^{(0)}).$$

Hence $\tilde{d}(x^{(k+1)}, z^{(k+1)}; x^{(k)}, z^{(k)}) \to 0$.

4.2.4 Convergence of iterates

Convergence of iterates can be shown under additional assumptions about the primal and dual distance functions. The following two assumptions are common in the literature on Bregman distances [CT93, Eck93, Gül94, CZ97].

1. For fixed x and z, the sublevel sets $\{x' \mid d_{\mathrm{p}}(x,x') \le \alpha\}$ and $\{z' \mid d_{\mathrm{d}}(z,z') \le \alpha\}$ are closed. In other words, the distances $d_{\mathrm{p}}(x,x')$ and $d_{\mathrm{d}}(z,z')$ are closed functions of x' and z', respectively. Since a sum of closed functions is closed, the distance $d(x,z;x',z')$ is a closed function of (x',z'), for fixed (x,z).

2. If $\tilde{x}^{(k)} \in \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{p}})$ converges to $x \in \mathbf{dom}\,\phi_{\mathrm{p}}$, then $d_{\mathrm{p}}(x,\tilde{x}^{(k)}) \to 0$. Similarly, if $\tilde{z}^{(k)} \in \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{d}})$ converges to $z \in \mathbf{dom}\,\phi_{\mathrm{d}}$, then $d_{\mathrm{d}}(z,\tilde{z}^{(k)}) \to 0$.

We also assume that $\sigma\tau\|A\|^2 + \tau L < 1$. As shown in Section 4.1.1 this implies that the kernel functions ϕ_{pcv} and ϕ_{dcv} are strongly convex and that

$$\tilde{d}(x,z;x',z') \ge \frac{\alpha}{2\tau}\|x-x'\|_{\mathrm{p}}^2 + \frac{\alpha}{2\sigma}\|z-z'\|_{\mathrm{d}}^2 \tag{47}$$

for some $\alpha > 0$. Similarly, $\sigma\tau\|A\|^2 < 1$ implies that

$$d(x,z;x',z') \ge \frac{\beta}{2\tau}\|x-x'\|_{\mathrm{p}}^2 + \frac{\beta}{2\sigma}\|z-z'\|_{\mathrm{d}}^2 \tag{48}$$

for some $\beta > 0$. Recall that $d = d_-$, $\tilde{d} = d_{\mathrm{pcv}}$ for the Bregman primal Condat–Vũ algorithm (27), and $d = d_+$, $\tilde{d} = d_{\mathrm{dcv}}$ for the Bregman dual Condat–Vũ algorithm.

Proof. We first note that $\tilde{d}(x^{(k+1)},z^{(k+1)};x^{(k)},z^{(k)}) \to 0$ and (47) imply that $x^{(k+1)} - x^{(k)} \to 0$ and $z^{(k+1)} - z^{(k)} \to 0$.

The inequality (46), together with (48), implies that the sequence $(x^{(k)},z^{(k)})$ is bounded. Let $(x^{(k_i)},z^{(k_i)})$ be a convergent subsequence of $(x^{(k)},z^{(k)})$ with limit point $(\hat{x},\hat{z})$. Since $x^{(k_i+1)} - x^{(k_i)} \to 0$ and $z^{(k_i+1)} - z^{(k_i)} \to 0$, the sequence $(x^{(k_i+1)},z^{(k_i+1)})$ also converges to $(\hat{x},\hat{z})$. We show that $(\hat{x},\hat{z})$ satisfies the optimality condition (4).

From (46), $d(x^\star,z^\star;x^{(k_i)},z^{(k_i)})$ is bounded. Since the sublevel sets $\{(x',z') \mid d(x^\star,z^\star;x',z') \le \alpha\}$ are closed subsets of $\mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{p}}) \cap \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{d}})$, the limit $(\hat{x},\hat{z}) \in \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{p}}) \cap \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{d}})$. The iterates in the subsequence satisfy

$$\nabla\phi_{\mathrm{pd}}(x^{(k_i)},z^{(k_i)}) - \nabla\phi_{\mathrm{pd}}(x^{(k_i+1)},z^{(k_i+1)}) + \begin{bmatrix} -A^T z^{(k_i+1)} \\ Ax^{(k_i+1)} \end{bmatrix} \in \begin{bmatrix} \partial f(x^{(k_i+1)}) + \nabla h(x^{(k_i+1)}) \\ \partial g^*(z^{(k_i+1)}) \end{bmatrix}, \tag{49}$$

where $\phi_{\mathrm{pd}} = \phi_{\mathrm{pcv}}$ in the Bregman primal Condat–Vũ algorithm and $\phi_{\mathrm{pd}} = \phi_{\mathrm{dcv}}$ in the Bregman dual Condat–Vũ algorithm. The left-hand side of (49) converges to $(-A^T\hat{z}, A\hat{x})$ because $\nabla\phi_{\mathrm{pd}}$ is continuous on $\mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{pd}})$. Since the operator on right-hand side of (49) is maximal monotone the limit point $(\hat{x},\hat{z})$ satisfies the optimality condition

$$\begin{bmatrix} -A^T\hat{z} \\ A\hat{x} \end{bmatrix} \in \begin{bmatrix} \partial f(\hat{x}) + \nabla h(\hat{x}) \\ \partial g^*(\hat{z}) \end{bmatrix}$$

(see [Bré73, page 27], [Tse00, Lemma 3.2]).

To show convergence of the entire sequence $(x^{(k)},z^{(k)})$, we substitute $(\hat{x},\hat{z})$ in (40):

$$\mathcal{L}(x^{(k+1)},\hat{z}) - \mathcal{L}(\hat{x},z^{(k+1)}) \le d(\hat{x},\hat{z};x^{(k)},z^{(k)}) - d(\hat{x},\hat{z};x^{(k+1)},z^{(k+1)}).$$

Since the left-hand side is nonnegative, we have $d(\hat{x},\hat{z};x^{(k)},z^{(k)}) \le d(\hat{x},\hat{z};x^{(k-1)},z^{(k-1)})$ for all $k \ge 1$. This further implies that

$$d(\hat{x},\hat{z};x^{(k)},z^{(k)}) \le d(\hat{x},\hat{z};x^{(k_i)},z^{(k_i)})$$

for all $k \ge k_i$. By the second additional assumption mentioned above, the right-hand side converges to zero. Then the left-hand side also converges to zero and, from (48) $x^{(k)} \to \hat{x}$ and $z^{(k)} \to \hat{z}$. □

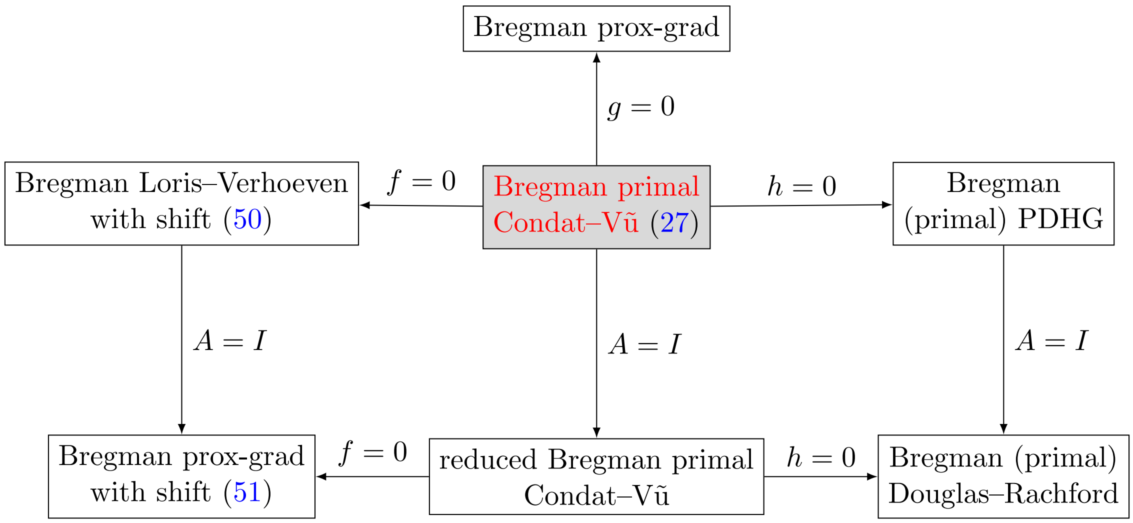

Figure 5: Summary of Bregman first-order proximal algorithms reduced from the Bregman primal Condat–Vũ algorithm (27).

4.3 Relation to other Bregman proximal splitting algorithms

Following similar steps as in Section 2, we obtain several Bregman proximal splitting methods as special cases of (27) and (28). The connections are summarized in Figure 5 and Figure 6. A comparison of Figures 1 and 5 shows that all the reduction relations ($A = I$) are still valid. However, it is unclear how to apply the "completion" operation to algorithms based on non-Euclidean Bregman distances.

When $h = 0$, (27) reduces to Bregman PDHG [CP16b]. When $g = 0$, $g^* = \delta_{\{0\}}$ (and assuming $z^{(0)} = 0$), we obtain the Bregman proximal gradient algorithm [BBT17]. When $f = 0$ in (27), we obtain the *Bregman Loris–Verhoeven algorithm with shift*:

$$x^{(k+1)} = \underset{x}{\operatorname{argmin}} \left(\langle \nabla h(x^{(k)}) - A^T z^{(k)}, x \rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)}) \right) \tag{50a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}} \left(z^{(k)}, -\sigma A(2x^{(k+1)} - x^{(k)}) \right). \tag{50b}$$

Furthermore, when $A = I$ in (27), we recover the reduced Bregman primal Condat–Vũ algorithm. Similarly, setting $A = I$ in Bregman PDHG yields the Bregman Douglas–Rachford algorithm. Last, when we set $A = I$ in (50), we have the *Bregman reduced Loris–Verhoeven algorithm with shift*:

$$x^{(k+1)} = \underset{x}{\operatorname{argmin}} \left(\langle \nabla h(x^{(k)}) - z^{(k)}, x \rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)}) \right) \tag{51a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}} \left(z^{(k)}, -\sigma(2x^{(k+1)} - x^{(k)}) \right). \tag{51b}$$

Similarly, the Bregman dual Condat–Vũ algorithm (28) can also be reduced to some other Bregman proximal splitting methods, as summarized in Figure 6. In particular, when $f = 0$ in (28), we obtain the *Bregman dual Loris–Verhoeven algorithm with shift*:

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}} (z^{(k)}, -\sigma A x^{(k)}) \tag{52a}$$

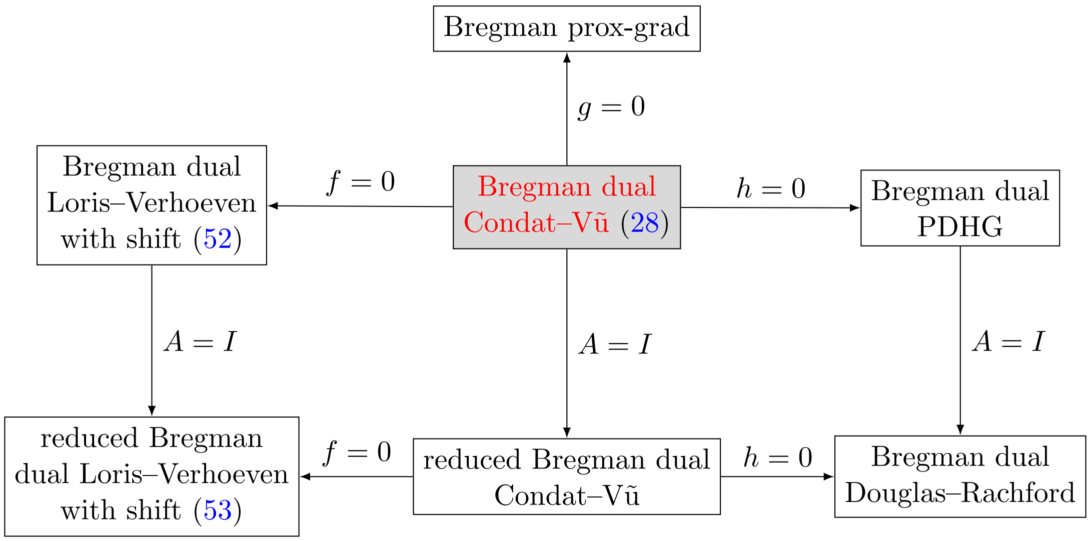

Figure 6: Summary of Bregman first-order proximal algorithms reduced from the Bregman dual Condat–Vũ algorithm (28).

$$x^{(k+1)} = \operatorname*{argmin}_{x} \big(\langle A^T(2z^{(k+1)} - z^{(k)}) + \nabla h(x^{(k)}), x\rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)})\big). \tag{52b}$$

Again we note that it is not (50) applied to the dual problem. Moreover, setting $A = I$ in (52) yields the *reduced Bregman Loris–Verhoeven algorithm with shift*:

$$z^{(k+1)} = \mathrm{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}}(z^{(k)}, -\sigma x^{(k)}) \tag{53a}$$

$$x^{(k+1)} = \operatorname*{argmin}_{x} \big(\langle 2z^{(k+1)} - z^{(k)} + \nabla h(x^{(k)}), x\rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)})\big). \tag{53b}$$

5 Bregman dual Condat–Vũ algorithm with line search

The proposed algorithms (27) and (28) use constant parameters σ and τ. The stepsize condition (38) involves the matrix norm $\|A\|$ and the Lipschitz constant L in (30). Estimating or bounding $\|A\|$ for a large matrix can be difficult, even in the Euclidean case. As an added complication, the norms $\|\cdot\|_{\mathrm{p}}$ and $\|\cdot\|_{\mathrm{d}}$ in the definition of the matrix norm (31) are assumed to be scaled so that the strong convexity parameters of the primal and dual kernels are equal to one. Close bounds on the strong convexity parameters may also be difficult to obtain. Using conservative bounds for $\|A\|$ and L may result in unnecessarily small values of σ and τ, and dramatically slow down the convergence. Even when the estimates of $\|A\|$ and L are accurate, the requirements for the stepsizes (38) are still too strict in most iterations, as observed in [ADH+21]. In view of the above arguments, line search techniques for primal–dual proximal methods have recently become an active area of research. Malitsky and Pock [MP18] proposed a line search technique for PDHG and the Condat–Vũ algorithm in the Euclidean case. The algorithm with adaptive parameters in [VMC21] focuses on a special case of (1) (*i.e.*, $f = 0$) and extends the Loris–Verhoeven algorithm (19). A Bregman proximal splitting method with line search is discussed in [JV21] and considers the problem (1) with $h = 0$ and $g = \delta_{\{b\}}$. In this section, we extend the Bregman dual Condat–Vũ algorithm (28) with a varying parameter option, in which the stepsizes are chosen adaptively without requiring any estimates or bounds for $\|A\|$ or the strong convexity parameter of the kernels. The algorithm is restricted to problems in the equality constrained form

$$\begin{array}{ll} \text{minimize} & f(x)+h(x) \\ \text{subject to} & Ax=b. \end{array} \tag{54}$$

This is a special case of (1) with $g = \delta_{\{b\}}$, the indicator function of the singleton $\{b\}$. While the standard form (1) is more general, one can note that methods for the equality constrained problem (54) also apply to (1) if it is reformulated as

$$\begin{array}{ll} \text{minimize} & f(x)+h(x)+g(y) \\ \text{subject to} & Ax-y=0. \end{array}$$

The details of the algorithm are discussed in Section 5.1 and a convergence analysis is presented in Section 5.2. The main conclusion is an $O(1/k)$ rate of ergodic convergence, consistent with previous results for related algorithms [MP18, JV21].

Assumptions We make the same assumptions as in Section 4.1, but define

$$\phi_{\mathrm{d}}(z) = \frac{1}{2}\|z\|, \qquad d_{\mathrm{d}}(z,z') = \frac{1}{2}\|z-z'\|^2, \qquad \|z\|_{\mathrm{d}} = \|z\|,$$

where $\|\cdot\|$ is Euclidean norm, and define $\|A\|$ correspondingly as

$$\|A\| = \sup_{u\neq 0, v\neq 0} \frac{\langle v, Au\rangle}{\|v\|\|u\|_{\mathrm{p}}} = \sup_{u\neq 0} \frac{\|Au\|}{\|u\|_{\mathrm{p}}} = \sup_{v\neq 0} \frac{\|A^T v\|_{\mathrm{p},*}}{\|v\|}.$$

5.1 Algorithm

The algorithm uses the following iteration, with starting points $x^{(0)} \in \mathbf{int}(\mathbf{dom}\,\phi_{\mathrm{p}}) \cap \mathbf{dom}\,h$ and $z^{(-1)} = z^{(0)}$:

$$\bar{z}^{(k+1)} = z^{(k)} + \theta_k(z^{(k)} - z^{(k-1)}) \tag{55a}$$
$$x^{(k+1)} = \mathrm{prox}_{\tau_k f}^{\phi_{\mathrm{p}}}\big(x^{(k)}, \tau_k(A^T\bar{z}^{(k+1)} + \nabla h(x^{(k)}))\big) \tag{55b}$$
$$z^{(k+1)} = z^{(k)} + \sigma_k(Ax^{(k+1)} - b). \tag{55c}$$

With constant parameters $\theta_k = 1$, $\sigma_k = \sigma$, $\tau_k = \tau$, the algorithm can be simplified as

$$\begin{aligned} x^{(k+1)} &= \mathrm{prox}_{\tau f}^{\phi_{\mathrm{p}}}\big(x^{(k)}, \tau A^T(2z^{(k)} - z^{(k-1)}) + \tau\nabla h(x^{(k)}))\big) \\ z^{(k+1)} &= z^{(k)} + \sigma(Ax^{(k+1)} - b). \end{aligned}$$

Except for the numbering of the dual iterates, this is the Bregman dual Condat–Vũ algorithm (28) applied to (54).

In the line search algorithm, the parameters θ_k, τ_k, σ_k are determined by a backtracking search. At the start of the algorithm, we set τ_{-1} and σ_{-1} to some positive values. To start the search in iteration k we choose $\bar{\theta}_k \geq 1$. For $i = 0, 1, 2, \ldots$, we set $\theta_k = 2^{-i}\bar{\theta}_k$, $\tau_k = \theta_k \tau_{k-1}$, $\sigma_k = \theta_k \sigma_{k-1}$, and compute $\bar{z}_{k+1}$, x_{k+1}, z_{k+1} using (55). For some $\delta \in (0, 1]$, if

$$\begin{aligned} & \langle z^{(k+1)} - \bar{z}^{(k+1)}, A(x^{(k+1)} - x^{(k)})\rangle + h(x^{(k+1)}) - h(x^{(k)}) - \langle \nabla h(x^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\ \leq \;& \frac{\delta^2}{\tau_k} d_{\mathrm{p}}(x^{(k+1)}, x^{(k)}) + \frac{1}{2\sigma_k}\|\bar{z}^{(k+1)} - z^{(k+1)}\|^2, \end{aligned} \tag{56}$$

we accept the computed iterates $\bar{z}^{(k+1)}$, $x^{(k+1)}$, $z^{(k+1)}$ and parameters θ_k, σ_k, τ_k, and terminate the backtracking search. If (56) does not hold, we increment i and continue the backtracking search.

The backtracking condition (56) is similar to the condition in the line search algorithm for PDHG with Euclidean proximal operators [MP18, Algorithm 4], but it is not identical, even in the Euclidean case. The proposed condition is weaker and allows larger stepsizes than the condition in [MP18, Algorithm 4].

5.2 Convergence analysis

The proof strategy is the same as in [JV21, Section 3.3], extended to account for the function h. The main conclusion is an $O(1/k)$ rate of ergodic convergence, shown in equation (64).

5.2.1 Lower bound on algorithm parameters

We first show that the stepsizes are bounded below by

$$\tau_k \geq \tau_{\min} \triangleq \min\Big\{\tau_{-1}, \frac{-L + \sqrt{L^2 + 4\delta^2\beta^2\|A\|^2}}{2\beta^2\|A\|^2}\Big\}, \qquad \sigma_k \geq \sigma_{\min} \triangleq \beta\tau_{\min}. \tag{57}$$

where $\beta = \sigma_{-1}/\tau_{-1}$. The lower bounds imply that the backtracking finally terminates and returns positive stepsizes σ_k and τ_k.

Proof. Applying the result in Section 4.1.1, with $\tau = \tau_k/\delta^2$, $\sigma = \sigma_k$, we see that the backtracking condition (56) holds at iteration k if

$$\tau_k\sigma_k\|A\|^2 + \tau_k L \leq \delta^2.$$

Then mathematical induction can be used to prove (57). The two lower bounds (57) hold at $k = 0$ by the definition of $\tau_{\min}$ and $\sigma_{\min}$. Now assume $\tau_{k-1} \geq \tau_{\min}$, $\sigma_{k-1} \geq \sigma_{\min}$, and consider the kth iteration. The first attempt of θ_k is $\theta_k = \bar{\theta}_k \geq 1$. If this value is accepted, then

$$\tau_k = \bar{\theta}_k\tau_{k-1} \geq \tau_{k-1} \geq \tau_{\min}, \qquad \sigma_k = \bar{\theta}_k\sigma_{k-1} \geq \sigma_{k-1} \geq \sigma_{\min}.$$

Otherwise, one or more backtracking steps are needed. Denote by $\tilde{\theta}_k$ the last rejected value. Then $\tilde{\theta}_k^2\tau_{k-1}^2\beta^2\|A\|^2 + \tilde{\theta}_k\tau_{k-1}L > \delta^2$ and the accepted θ_k satisfies

$$\theta_k = \frac{\tilde{\theta}_k}{2} \geq \frac{-L + \sqrt{L^2 + 4\delta^2\beta^2\|A\|^2}}{2\tau_{k-1}\beta^2\|A\|^2}.$$

Therefore,

$$\tau_k = \theta_k\tau_{k-1} > \frac{-L + \sqrt{L^2 + 4\delta^2\beta^2\|A\|^2}}{2\beta^2\|A\|^2}, \qquad \sigma_k = \beta\tau_k \geq \beta\tau_{\min}. \qquad \square$$

5.2.2 One-iteration analysis

The iterates $x^{(k+1)}$, $z^{(k+1)}$, $\bar{z}^{(k+1)}$ generated by the algorithm (55) satisfy

$$\begin{aligned}\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, \bar{z}^{(k+1)}) \quad &\leq \quad \frac{1}{\tau_k}\left(d_{\mathrm{p}}(x, x^{(k)}) - d_{\mathrm{p}}(x, x^{(k+1)}) - (1-\delta^2) d_{\mathrm{p}}(x^{(k+1)}, x^{(k)})\right) \\ &\quad + \frac{1}{2\sigma_k}\left(\|z - z^{(k)}\|^2 - \|z - z^{(k+1)}\|^2 - \|\bar{z}^{(k+1)} - z^{(k)}\|^2\right) \qquad (58)\end{aligned}$$

for all $x \in \mathbf{dom}\, f \cap \mathbf{dom}\, \phi_{\mathrm{p}}$ and all z. Here $\mathcal{L}(x, z) = f(x) + h(x) + \langle z, Ax - b\rangle$.

Proof. The optimality condition for the primal prox-operator (55b) gives

$$\begin{aligned}& f(x^{(k+1)}) - f(x) \\ &\leq \quad \frac{1}{\tau_k}\big(d_{\mathrm{p}}(x, x^{(k)}) - d_{\mathrm{p}}(x, x^{(k+1)}) - d_{\mathrm{p}}(x^{(k+1)}, x^{(k)})\big) + \langle A^T \bar{z}^{(k+1)} + \nabla h(x^{(k)}), x - x^{(k+1)}\rangle,\end{aligned}$$

and hence

$$\begin{aligned}& f(x^{(k+1)}) + h(x^{(k+1)}) - f(x) - h(x) \\ &\leq \quad \frac{1}{\tau_k}(d_{\mathrm{p}}(x, x^{(k)}) - d_{\mathrm{p}}(x, x^{(k+1)}) - d_{\mathrm{p}}(x^{(k+1)}, x^{(k)})) + \langle A^T \bar{z}_{k+1}, x - x^{(k+1)}\rangle \\ &\qquad + h(x^{(k+1)}) - h(x) + \langle \nabla h(x^{(k)}), x - x^{(k+1)}\rangle \\ &\leq \quad \frac{1}{\tau_k}(d_{\mathrm{p}}(x, x^{(k)}) - d_{\mathrm{p}}(x, x^{(k+1)}) - d_{\mathrm{p}}(x^{(k+1)}, x^{(k)})) + \langle A^T \bar{z}^{(k+1)}, x - x^{(k+1)}\rangle \\ &\qquad + h(x^{(k+1)}) - h(x^{(k)}) - \langle \nabla h(x^{(k)}), x^{(k+1)} - x^{(k)}\rangle. \qquad (59)\end{aligned}$$

The second inequality follows from the convexity of h, *i.e.*, $h(x) \geq h(x^{(k)}) + \langle \nabla h(x^{(k)}), x - x^{(k)}\rangle$. The dual update (55c) implies that

$$\langle z - z^{(k+1)}, Ax^{(k+1)} - b\rangle = \frac{1}{\sigma_k}\langle z - z^{(k+1)}, z^{(k+1)} - z^{(k)}\rangle \quad \text{for all } z. \qquad (60)$$

This equality at $k = i - 1$ is

$$\begin{aligned}\langle z - z^{(i)}, Ax^{(i)} - b\rangle \quad &= \quad \frac{1}{\sigma_{i-1}}\langle z - z^{(i)}, z^{(i)} - z^{(i-1)}\rangle \\ &= \quad \frac{1}{2\sigma_{i-1}}\left(\|z - z^{(i-1)}\|^2 - \|z - z^{(i)}\|^2 - \|z^{(i)} - z^{(i-1)}\|^2\right). \qquad (61)\end{aligned}$$

The equality (60) at $k = i - 2$ is

$$\begin{aligned}\langle z - z^{(i-1)}, Ax^{(i-1)} - b\rangle \quad &= \quad \frac{1}{\sigma_{i-2}}\langle z - z^{(i-1)}, z^{(i-1)} - z^{(i-2)}\rangle \\ &= \quad \frac{\theta_{i-1}}{\sigma_{i-1}}\langle z - z^{(i-1)}, z^{(i-1)} - z^{(i-2)}\rangle \\ &= \quad \frac{1}{\sigma_{i-1}}\langle z - z^{(i-1)}, \bar{z}^{(i)} - z^{(i-1)}\rangle.\end{aligned}$$

We evaluate this at $z = z^{(i)}$ and add it to the inequality at $z = z^{(i-2)}$ multiplied by θ_{i-1}:

$$\begin{aligned}
\langle z^{(i)} - \bar{z}^{(i)}, Ax^{(i-1)} - b\rangle &= \frac{1}{\sigma_{i-1}}\langle z^{(i)} - \bar{z}^{(i)}, \bar{z}^{(i)} - z^{(i-1)}\rangle \\
&= \frac{1}{2\sigma_{i-1}}\left(\|z^{(i)} - z^{(i-1)}\|^2 - \|z^{(i)} - \bar{z}^{(i)}\|^2 - \|\bar{z}^{(i)} - z^{(i-1)}\|^2\right). \qquad (62)
\end{aligned}$$

Now we combine (59) for $k = i-1$, with (61) and (62). For $i \geq 1$,

$$\begin{aligned}
&\mathcal{L}(x^{(i)}, z) - \mathcal{L}(x, \bar{z}^{(i)}) \\
&= f(x^{(i)}) + h(x^{(i)}) + \langle z, Ax^{(i)} - b\rangle - f(x) - h(x) - \langle \bar{z}^{(i)}, Ax - b\rangle \\
&\leq \frac{1}{\tau_{i-1}}\Big(d_{\mathrm{p}}(x, x^{(i-1)}) - d_{\mathrm{p}}(x, x^{(i)}) - d_{\mathrm{p}}(x^{(i)}, x^{(i-1)})\Big) + \langle A^T\bar{z}^{(i)}, x - x^{(i)}\rangle + \langle z, Ax^{(i)} - b\rangle \\
&\quad - \langle \bar{z}^{(i)}, Ax - b\rangle + h(x^{(i)}) - h(x^{(i-1)}) - \langle \nabla h(x^{(i-1)}), x^{(i)} - x^{(i-1)}\rangle \\
&= \frac{1}{\tau_{i-1}}\Big(d_{\mathrm{p}}(x, x^{(i-1)}) - d_{\mathrm{p}}(x, x^{(i)}) - d_{\mathrm{p}}(x^{(i)}, x^{(i-1)})\Big) + \langle z - \bar{z}^{(i)}, Ax^{(i)} - b\rangle \\
&\quad + h(x^{(i)}) - h(x^{(i-1)}) - \langle \nabla h(x^{(i-1)}), x^{(i)} - x^{(i-1)}\rangle \\
&= \frac{1}{\tau_{i-1}}\Big(d_{\mathrm{p}}(x, x^{(i-1)}) - d_{\mathrm{p}}(x, x^{(i)}) - d_{\mathrm{p}}(x^{(i)}, x^{(i-1)})\Big) \\
&\quad + \langle z^{(i)} - \bar{z}^{(i)}, A(x^{(i)} - x^{(i-1)})\rangle + \langle z - z^{(i)}, Ax^{(i)} - b\rangle + \langle z^{(i)} - \bar{z}^{(i)}, Ax^{(i-1)} - b\rangle \\
&\quad + h(x^{(i)}) - h(x^{(i-1)}) - \langle \nabla h(x^{(i-1)}), x^{(i)} - x^{(i-1)}\rangle \\
&= \frac{1}{\tau_{i-1}}\Big(d_{\mathrm{p}}(x, x^{(i-1)}) - d_{\mathrm{p}}(x, x^{(i)}) - d_{\mathrm{p}}(x^{(i)}, x^{(i-1)}))\Big) \\
&\quad + \frac{1}{2\sigma_{i-1}}\Big(\|z - z^{(i-1)}\|^2 - \|z - z^{(i)}\|^2 - \|\bar{z}^{(i)} - z^{(i-1)}\|^2 - \|\bar{z}^{(i)} - z^{(i)}\|^2\Big) \\
&\quad + \langle A^T(z^{(i)} - \bar{z}^{(i)}), x^{(i)} - x^{(i-1)}\rangle + h(x^{(i)}) - h(x^{(i-1)}) - \langle \nabla h(x^{(i-1)}), x^{(i)} - x^{(i-1)})\rangle \\
&\leq \frac{1}{\tau_{i-1}}\left(d_{\mathrm{p}}(x, x^{(i-1)}) - d_{\mathrm{p}}(x, x^{(i)}) - (1 - \delta^2)d_{\mathrm{p}}(x^{(i)}, x^{(i-1)})\right) \\
&\quad + \frac{1}{2\sigma_{i-1}}\left(\|z - z^{(i-1)}\|^2 - \|z - z^{(i)}\|^2 - \|\bar{z}^{(i)} - z^{(i-1)}\|^2\right),
\end{aligned}$$

which is the desired result (58). The first inequality follows from (59). In the second last step we substitute (61) and (62). The last step uses the line search exit condition (56) at $k = i-1$. $\square$

5.2.3 Ergodic convergence

We define the averaged primal and dual sequences

$$x_{\mathrm{avg}}^{(k)} = \frac{1}{\sum_{i=1}^k \tau_{i-1}}\sum_{i=1}^k \tau_{i-1}x^{(i)}, \qquad \bar{z}_{\mathrm{avg}}^{(k)} = \frac{1}{\sum_{i=1}^k \tau_{i-1}}\sum_{i=1}^k \tau_{i-1}\bar{z}^{(i)}$$

for $k \geq 1$. We show that

$$\mathcal{L}(x_{\mathrm{avg}}^{(k)}, z^\star) - \mathcal{L}(x^\star, \bar{z}_{\mathrm{avg}}^{(k)}) \leq \frac{1}{\sum_{i=1}^k \tau_{i-1}}\big(d(x^\star, x^{(0)}) + \frac{1}{2\beta}\|z^\star - z^{(0)}\|^2\big) \qquad (63)$$

$$\leq \quad \frac{1}{k\tau_{\min}}\big(d(x^\star, x^{(0)}) + \frac{1}{2\beta}\|z^\star - z^{(0)}\|^2\big). \tag{64}$$

This holds for any choice of $\delta \in (0, 1]$ in (56). If we compare this with (44), we note that the left-hand side involves a different dual iterate ($\bar{z}^{(k)}_{\text{avg}}$ as opposed to $z^{(k)}_{\text{avg}}$).

Proof. From (58),

$$\mathcal{L}(x^{(i)}, z) - \mathcal{L}(x, \bar{z}^{(i)}) \leq \frac{1}{\tau_{i-1}}\Big(d_{\text{p}}(x, x^{(i-1)}) - d_{\text{p}}(x, x^{(i)}) + \frac{1}{2\beta}\|z - z^{(i-1)}\|^2 - \frac{1}{2\beta}\|z - z^{(i)}\|^2\Big).$$

Since $\mathcal{L}$ is convex in x and affine in z,

$$\begin{aligned}
\big(\sum_{i=1}^{k}\tau_{i-1}\big)\big(\mathcal{L}(x^{(k)}_{\text{avg}}, z) - \mathcal{L}(x, \bar{z}^{(k)}_{\text{avg}})\big) \quad &\leq \quad \sum_{i=1}^{k}\tau_{i-1}(\mathcal{L}(x^{(i)}, z) - \mathcal{L}(x, \bar{z}^{(i)})) \\
&\leq \quad d_{\text{p}}(x, x^{(0)}) - d_{\text{p}}(x, x^{(k)}) + \frac{1}{2\beta}(\|z - z^{(0)}\|^2 - \|z - z^{(k)}\|^2) \qquad (65) \\
&\leq \quad d_{\text{p}}(x, x^{(0)}) + \frac{1}{2\beta}\|z - z^{(0)}\|^2. \qquad (66)
\end{aligned}$$

Dividing by $\sum_{i=1}^{k}\tau_{i-1}$ and plugging in $x = x^\star$, $z = z^\star$ gives (63). ☐

Substituting $x = x^\star$ in (66) yields

$$f(x^{(k)}_{\text{avg}}) + z^T(Ax^{(k)}_{\text{avg}} - b) - f(x^\star) \leq \frac{1}{\sum_{i=1}^{k}\tau_{i-1}}\big(d_{\text{p}}(x^\star, x^{(0)}) + \frac{1}{2\beta}\|z - z^{(0)}\|^2\big) \quad \text{for any } z,$$

since $Ax^\star = b$. The inequality still holds if we maximize both sides over z subject to $\|z\| \leq \gamma$:

$$f(x^{(k)}_{\text{avg}}) + h(x^{(k)}_{\text{avg}}) + \gamma\|Ax^{(k)}_{\text{avg}} - b\| - f(x^\star) - h(x^\star) \leq \frac{1}{\sum_{i=1}^{k}\tau_{i-1}}\Big(d_{\text{p}}(x^\star, x^{(0)}) + \frac{1}{2\beta}(\gamma + \|z^{(0)}\|)^2\Big).$$

The function $f(x) + h(x) + \gamma\|Ax - b\|$ on the left-hand side is called the *merit function* [ST14, page 287]. For $\gamma > \|z^\star\|$, the penalty $\gamma\|Ax - b\|$ in the merit function is *exact*, *i.e.*, $f(x) + h(x) + \gamma\|Ax - b\| - f(x^\star) - h(x^\star) \geq 0$ with equality only if x is optimal. Since $\tau_i \geq \tau_{\min}$, the inequality shows that the merit function decreases as $O(1/k)$.

5.2.4 Monotonicity properties and convergence of iterates

For $x = x^\star$, $z = z^\star$, the left-hand side of (58) is nonnegative and we obtain

$$\begin{aligned}
& d(x^\star, x^{(k+1)}) + \frac{1}{2\beta}\|z^\star - z^{(k+1)}\|^2 \\
& \quad \leq \quad d(x^\star, x^{(k)}) + \frac{1}{2\beta}\|z^\star - z^{(k)}\|^2 - \big((1 - \delta^2)d(x^\star, x^{(k)}) + \frac{1}{2\beta}\|z^\star - z^{(k)}\|^2\big) \\
& \quad \leq \quad d(x^\star, x^{(k)}) + \frac{1}{2\beta}\|z^\star - z^{(k)}\|^2
\end{aligned}$$

for $k \geq 0$. Moreover,

$$\sum_{i=0}^{k} \left((1-\delta^2)(d_{\mathrm{p}}(x^{(i+1)}, x^{(i)}) + \frac{1}{2\beta} \|\bar{z}^{(i+1)} - z^{(i)}\|^2 \right) \leq d_{\mathrm{p}}(x^\star, x^{(0)}) + \frac{1}{2\beta} \|z^\star - \bar{z}^{(0)}\|^2.$$

These inequalities hold for any value $\delta \in (0, 1]$. In particular, the last inequality implies that $\bar{z}^{(i+1)} - z^{(i)} \to 0$. When $\delta < 1$ it also implies that $d_{\mathrm{p}}(x^{(i+1)}, x^{(i)}) \to 0$ and, by the strong convexity assumption on ϕ_{p}, that $x^{(i+1)} - x^{(i)} \to 0$. With additional assumptions similar to those in Section 4.2.3, one can also show the convergence of iterates; see [JV21, Section 3.3.4].

6 Bregman PD3O algorithm

In this section we propose the *Bregman PD3O algorithm*, another Bregman proximal method for the problem (1). Bregman PD3O also involves two generalized distances, d_{p} and d_{d}, generated by ϕ_{p} and ϕ_{d}, respectively, and it consists of the iterations

$$x^{(k+1)} = \operatorname{prox}_{\tau f}^{\phi_{\mathrm{p}}}(x^{(k)}, \tau A^T z^{(k)} + \tau \nabla h(x^{(k)})) \tag{67a}$$

$$z^{(k+1)} = \operatorname{prox}_{\sigma g^*}^{\phi_{\mathrm{d}}}(z^{(k)}, -\sigma A(2x^{(k+1)} - x^{(k)} + \tau(\nabla h(x^{(k)}) - \nabla h(x^{(k+1)})))). \tag{67b}$$

The only difference between Bregman PD3O and Bregman primal Condat–Vũ algorithm (27) is the additional term $\tau(\nabla h(x^{(k)}) - \nabla h(x^{(k+1)}))$. Thus the two algorithms (27) and (67) reduce to the same method when h is absent from problem (1). In the Eucliden case, the additional term allows PD3O to admit larger parameters compared with the Condat–Vũ algorithm:

$$\begin{array}{ll} \text{Condat–Vũ} & \sigma\tau\|A\|^2 + \tau L \leq 1 \\ \text{PD3O} & \sigma\tau\|A\|^2 \leq 1, \quad \tau \leq 1/L. \end{array} \tag{68}$$

The range of possible parameters is illustrated in Figure 7. When Bregman distances are used, however, the definitions of the matrix norm $\|A\|$ and the Lipschitz constant L are different in Bregman PD3O and Bregman Condat–Vũ algorithms, due to the limitation of our analysis in Section 6.1. Thus, Bregman PD3O does not necessarily admit larger parameters than Bregman Condat–Vũ algorithms.

In Section 6.1 we provide the detailed convergence analysis. The connections between Bregman PD3O and several other Bregman proximal methods are discussed in Section 6.2.

Assumptions Throughout Section 6 we make the following assumptions. The kernel functions ϕ_{p} and ϕ_{d} are 1-strongly convex with respect to the Euclidean norm and an arbitrary norm $\|\cdot\|_{\mathrm{d}}$, respectively:

$$d_{\mathrm{p}}(x, x') \geq \frac{1}{2}\|x - x'\|^2, \qquad d_{\mathrm{d}}(z, z') \geq \frac{1}{2}\|z - z'\|_{\mathrm{d}}^2. \tag{69}$$

The assumptions that the strong convexity constants are one can be made without loss of generality, by scaling the distances. The definition of $\|A\|$ follows (31) and reduces to

$$\|A\| = \sup_{v \neq 0} \frac{\|Av\|_{\mathrm{d},*}}{\|v\|}.$$

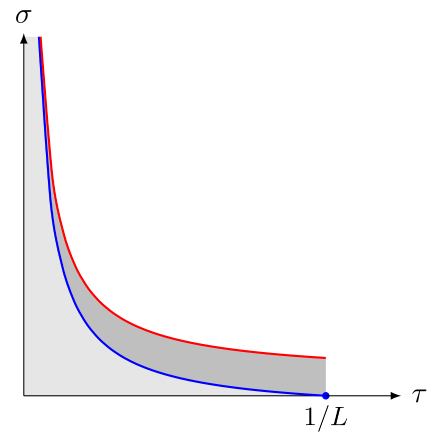

Figure 7: Illustration of choices of parameters in Euclidean Condat–Vũ algorithms and PD3O. The light gray region under the blue curve shows the possible choices for (τ, σ) in Condat–Vũ algorithms. The light gray and dark gray regions together shows the possible choices for (τ, σ) in PD3O. In the Euclidean case, PD3O admits a larger choice of parameters.

We also assume that the gradient of h is L-Lipschitz continuous with respect to Euclidean norm: $\mathbf{dom}\, h = \mathbf{R}^n$ and

$$h(y) - h(x) - \langle \nabla h(x), y - x \rangle \leq \frac{L}{2}\|y - x\|^2, \quad \text{for any } x, y \in \mathbf{dom}\, h. \tag{70}$$

The parameters τ and σ must satisfy

$$\sigma\tau\|A\|^2 \leq 1, \qquad \tau \leq 1/L. \tag{71}$$

Finally, we assume that the optimality condition (4) has a solution $(x^\star, z^\star) \in \mathbf{dom}\, \phi_{\mathrm{p}} \times \mathbf{dom}\, \phi_{\mathrm{d}}$.

Note that (70) is a stronger assumption than (30). It implies (30) if we combine it with the first inequality in (69). We will use the following consequence of (70):

$$h(y) - h(x) - \langle \nabla h(x), y - x \rangle \geq \frac{1}{2L}\|\nabla h(y) - \nabla h(x)\|^2 \tag{72}$$

for all x, y [Nes18, Theorem 2.1.5].

6.1 Convergence analysis

6.1.1 A primal–dual Bregman distance

We introduce a primal–dual kernel

$$\phi_{\mathrm{pd3o}}(x, y, z) = \frac{1}{\tau}\phi_{\mathrm{p}}(x) + \frac{1}{\sigma}\phi_{\mathrm{d}}(z) + \frac{\tau}{2}\|y\|^2 - \langle y, x \rangle - \langle z, A(x - \tau y) \rangle,$$

where $\sigma, \tau > 0$. If ϕ_{pd3o} is convex, the generated Bregman distance is given by

$$\begin{aligned} d_{\mathrm{pd3o}}(x, y, z; x', y', z') \;&=\; \frac{1}{\tau}d_{\mathrm{p}}(x, x') + \frac{1}{\sigma}d_{\mathrm{d}}(z, z') + \frac{\tau}{2}\|y - y'\|^2 \\ &\quad - \langle y - y', x - x' \rangle - \langle z - z', A(x - x') \rangle + \tau\langle z - z', A(y - y') \rangle \end{aligned} \tag{73}$$

where d_- is defined in (34). We now show that ϕ_{pd3o} is convex if $\sigma\tau\|A\|^2 \leq 1$.

Proof. It is sufficient to show that d_{pd3o} is nonnegative:

$$\begin{aligned} d_{\mathrm{pd3o}}(x, y, z; x', y', z') &\geq \frac{1}{2\tau}\|x - x'\|^2 + \frac{\tau}{2}\|A^T(z - z')\|^2 + \frac{\tau}{2}\|y - y'\|^2 \\ &\quad - \langle y - y', x - x'\rangle - \langle z - z', A(x - x')\rangle + \tau\langle z - z', A(y - y')\rangle \\ &= \frac{1}{2}\Big\|\frac{1}{\sqrt{\tau}}(x - x') - \sqrt{\tau}(y - y') - \sqrt{\tau}A^T(z - z')\Big\|^2 \\ &\geq 0. \end{aligned} \tag{74}$$

In step 1 we use the strong convexity assumption (29), the definition of $\|A\|$ (31), and the assumption $\sigma\tau\|A\|^2 \leq 1$. Especially we have

$$\frac{1}{\sigma}d_{\mathrm{d}}(z, z') \geq \frac{1}{2\sigma}\|z - z'\|_{\mathrm{d}}^2 \geq \frac{\|A^T(z - z')\|^2}{2\sigma\|A\|^2} \geq \frac{\tau}{2}\|A^T(z - z')\|^2. \qquad \square$$

Note that the convexity of ϕ_{pd3o} only requires the first inequality in the stepsize condition (71). Although the Bregman PD3O algorithm (67) is not the Bregman proximal point method for the Bregman kernel ϕ_{pd3o}, the distance d_{pd3o} will appear in the key inequality (75) of the convergence analysis.

6.1.2 One-iteration analysis

We first show that the iterates $x^{(k+1)}$, $z^{(k+1)}$ generated by Bregman PD3O (67) satisfy

$$\begin{aligned} &\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\ &\quad\leq d_{\mathrm{pd3o}}\big(x, \nabla h(x), z; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}\big) - d_{\mathrm{pd3o}}\big(x, \nabla h(x), z; x^{(k+1)}, \nabla h(x^{(k+1)}), z^{(k+1)}\big) \\ &\qquad - d_{\mathrm{pd3o}}\big(x^{(k+1)}, \nabla h(x), z^{(k+1)}; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}\big) \end{aligned} \tag{75}$$

for all $x \in \mathbf{dom}\, f \cap \mathbf{dom}\, \phi_{\mathrm{p}}$ and $z \in \mathbf{dom}\, g^* \cap \mathbf{dom}\, \phi_{\mathrm{d}}$.

Proof. Recall that Bregman PD3O differs from the Bregman primal Condat–Vũ algorithm (27) only in an additional term in the dual update. The proof in Section 5.2.2 therefore applies up to (42), with

$$\tilde{x} = 2x^{(k+1)} - x^{(k)} + \tau(\nabla h(x^{(k)}) - \nabla h(x^{(k+1)})), \qquad \tilde{z} = z^{(k)}.$$

Hence,

$$\begin{aligned} &\mathcal{L}(x^{(k+1)}, z) - \mathcal{L}(x, z^{(k+1)}) \\ &\quad\leq d_{-}(x, z; x^{(k)}, z^{(k)}) - d_{-}(x, z; x^{(k+1)}, z^{(k+1)}) - d_{-}(x^{(k+1)}, z^{(k+1)}; x^{(k)}, z^{(k)}) \\ &\qquad - \tau\langle A^T(z - z^{(k+1)}), \nabla h(x^{(k)}) - \nabla h(x^{(k+1)})\rangle \\ &\qquad - h(x^{(k+1)}) + h(x^{(k)}) + \langle \nabla h(x^{(k)}), x^{(k+1)} - x^{(k)}\rangle \\ &\quad= d_{-}(x, z; x^{(k)}, z^{(k)}) + \frac{\tau}{2}\|\nabla h(x) - \nabla h(x^{(k)})\|^2 \\ &\qquad - \langle (x - \tau A^T z) - (x^{(k)} - \tau A^T z^{(k)}), \nabla h(x) - \nabla h(x^{(k)})\rangle \\ &\qquad - \Big(d_{-}(x, z; x^{(k+1)}, z^{(k+1)}) + \frac{\tau}{2}\|\nabla h(x) - \nabla h(x^{(k+1)})\|^2 \\ &\qquad - \big\langle x - \tau A^T z - (x^{(k+1)} - \tau A^T z^{(k+1)}), \nabla h(x) - \nabla h(x^{(k+1)})\big\rangle\Big) \end{aligned}$$

$$
\begin{aligned}
& - \Big(d_-(x^{(k+1)}, z^{(k+1)}; x^{(k)}, z^{(k)}) + \frac{\tau}{2}\|\nabla h(x) - \nabla h(x^{(k)})\|^2 \\
& - \big\langle (x^{(k+1)} - \tau A^T z^{(k+1)}) - (x^{(k)} - \tau A^T z^{(k)}), \nabla h(x) - \nabla h(x^{(k)}) \big\rangle \Big) \\
& - (h(x) - h(x^{(k+1)}) - \langle \nabla h(x^{(k+1)}), x - x^{(k+1)} \rangle - \frac{\tau}{2}\|\nabla h(x) - \nabla h(x^{(k+1)})\|^2) \\
= \;& d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}) - d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k+1)}, \nabla h(x^{(k+1)}), z^{(k+1)}) \\
& - d_{\mathrm{pd3o}}(x^{(k+1)}, \nabla h(x), z^{(k+1)}; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}) \\
& - (h(x) - h(x^{(k+1)}) - \langle \nabla h(x^{(k+1)}), x - x^{(k+1)} \rangle - \frac{\tau}{2}\|\nabla h(x) - \nabla h(x^{(k+1)})\|_{\mathrm{p},*}^2) \\
\le \;& d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}) - d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k+1)}, \nabla h(x^{(k+1)}), z^{(k+1)}) \\
& - d_{\mathrm{pd3o}}(x^{(k+1)}, \nabla h(x), z^{(k+1)}; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}) \\
\le \;& d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k)}, \nabla h(x^{(k)}), z^{(k)}) - d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k+1)}, \nabla h(x^{(k+1)}), z^{(k+1)}).
\end{aligned}
$$

Step 3 follows from definition of d_{ppd} (73). In step 4 we use the Lipschitz condition (72) and the second inequality in the stepsize condition (71). The last step follows from the fact that d_{pd3o} is nonnegative (74). □

6.1.3 Ergodic convergence

We show that the iterates generated by Bregman PD3O (67) satisfy

$$
\mathcal{L}(x_{\mathrm{avg}}^{(k)}, z^\star) - \mathcal{L}(x^\star, z_{\mathrm{avg}}^{(k)}) \le \frac{2}{k}\Big(\frac{1}{\tau} d_{\mathrm{p}}(x^\star, x^{(0)}) + \frac{1}{\sigma} d_{\mathrm{d}}(z^\star, z^{(0)}) + \frac{\tau}{2}\|\nabla h(x^\star) - \nabla h(x^{(0)})\|^2\Big), \tag{76}
$$

for $k \ge 1$, where the averaged iterates are defined in (43).

Proof. From (75), since $\mathcal{L}(u, v)$ is convex in u and concave in v,

$$
\begin{aligned}
& \mathcal{L}(x_{\mathrm{avg}}^{(k)}, z) - \mathcal{L}(x, z_{\mathrm{avg}}^{(k)}) \\
& \quad \le \; \frac{1}{k}\sum_{i=1}^{k} \big(\mathcal{L}(x^{(i)}, z) - \mathcal{L}(x, z^{(i)})\big) \\
& \quad \le \; \frac{1}{k}\Big(d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(0)}, \nabla h(x^{(0)}), z^{(0)}) - d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(k)}, \nabla h(x^{(k)}), z^{(k)})\Big) \\
& \quad \le \; \frac{1}{k} d_{\mathrm{pd3o}}(x, \nabla h(x), z; x^{(0)}, \nabla h(x^{(0)}), z^{(0)}) \\
& \quad \le \; \frac{2}{k}\Big(\frac{1}{\tau} d_{\mathrm{p}}(x, x^{(0)}) + \frac{1}{\sigma} d_{\mathrm{d}}(z, z^{(0)}) + \frac{\tau}{2}\|\nabla h(x) - \nabla h(x^{(0)})\|^2\Big)
\end{aligned}
$$

for all $x \in \mathbf{dom}\, f \cap \mathbf{dom}\, \phi_{\mathrm{p}}$ and $z \in \mathbf{dom}\, g \cap \mathbf{dom}\, \phi_{\mathrm{d}}$. Plugging in $x = x^\star$ and $z = z^\star$ yields the desired result (76). □

6.2 Relation to other Bregman proximal algorithms

The proposed algorithm (67) can be viewed as an extension to PD3O (18) using generalized distances, and reduces to several Bregman proximal methods by reduction. These algorithms can also be organized into a diagram similar to Figure 3. Figure 8 starts from Bregman PD3O (67), and

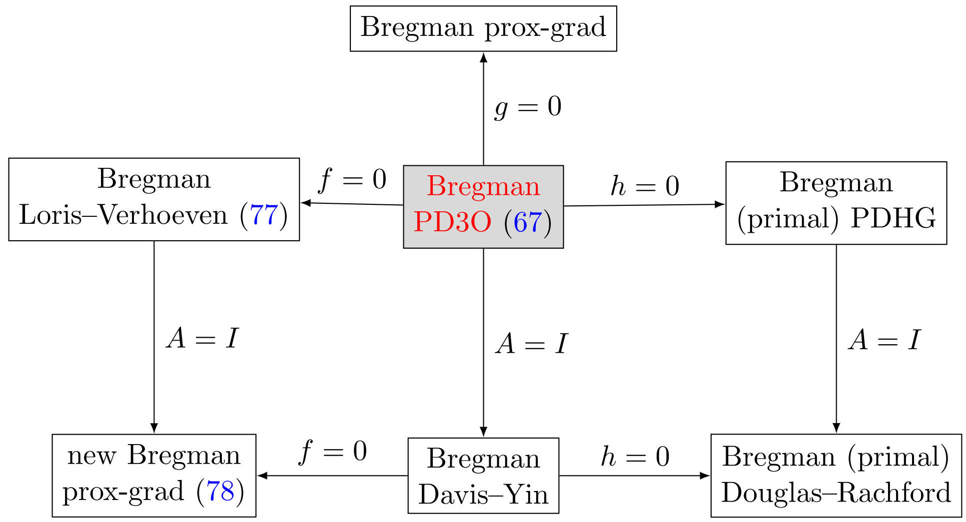

Figure 8: Summary of Bregman proximal algorithms reduced from Bregman PD3O algorithm.

summarizes its connection to several Bregman proximal methods. When $h = 0$, (67) reduces to Bregman PDHG, and when $g = 0$, (67) reduces to the Bregman proximal gradient algorithm. More interestingly, when $f = 0$, Bregman PD3O reduces to the Bregman Loris–Verhoeven algorithm

$$x^{(k+1)} = \operatorname*{argmin}_x \big(\langle \nabla h(x^{(k)}) - A^T z^{(k)}, x\rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)})\big) \tag{77a}$$

$$z^{(k+1)} = \operatorname{prox}^{\phi_{\mathrm{d}}}_{\sigma g^*}\big(z^{(k)}, -\sigma A\big(2x^{(k+1)} - x^{(k)} + \tau\big(\nabla h(x^{(k)}) - \nabla h(x^{(k+1)})\big)\big)\big). \tag{77b}$$

Setting $A = I$ (with $\sigma = 1/\tau$), we obtain a new variant of Bregman proximal gradient algorithm:

$$x^{(k+1)} = \operatorname*{argmin}_x \big(\langle \nabla h(x^{(k)}) - z^{(k)}, x\rangle + \frac{1}{\tau} d_{\mathrm{p}}(x, x^{(k)})\big) \tag{78a}$$

$$z^{(k+1)} = \operatorname{prox}^{\phi_{\mathrm{d}}}_{\tau^{-1} g^*}\big(z^{(k)}, -\frac{1}{\tau} A\big(2x^{(k+1)} - x^{(k)}\big) - A(\nabla h(x^{(k)}) - \nabla h(x^{(k+1)}))\big). \tag{78b}$$

The distinction between (78) and (51) is the additional term $\tau(\nabla h(x^{(k)} - \nabla h(x^{(k+1)})$, the same as the difference between (27) and (67). When the Euclidean proximal operator is used, (78) reduces to the proximal gradient method. Unfortunately, however, the new algorithm (78) does not seem to be equivalent to the Bregman proximal gradient algorithm due to the lack of Moreau decomposition in the generalized case. Nevertheless, the new algorithm (78) is still interesting on its own, especially when the generalized proximal operator of g^* is easy to compute while the (Euclidean and generalized) proximal operator of g is computationally expensive. Finally, setting $A = I$ (and $\sigma = 1/\tau$) in Bregman PD3O (67) gives a Bregman Davis–Yin algorithm.

7 Numerical experiment

In this section we evaluate the performance of the Bregman primal Condat–Vũ algorithm (27), Bregman dual Condat–Vũ algorithm with line search (55), and Bregman PD3O (67). The main goal of the example is to validate and illustrate the difference in the stepsize conditions (68), and the usefulness of the line search procedure. We consider the convex optimization problem

$$\begin{array}{ll} \text{minimize} & \psi(x) = \lambda\|Ax\|_1 + \frac{1}{2}\|Cx-b\|^2 \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0, \end{array} \tag{79}$$

where $x \in \mathbf{R}^n$ is the optimization variable, $C \in \mathbf{R}^{m\times n}$, and $A \in \mathbf{R}^{(n-1)\times n}$ is the difference matrix

$$A = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}. \tag{80}$$

This problem is of the form of (1) with

$$f(x) = \delta_H(x), \qquad g(y) = \lambda\|y\|_1, \qquad g^*(z) = \begin{cases} 0 & \|z\|_\infty \le \lambda \\ +\infty & \text{otherwise}, \end{cases} \qquad h(x) = \frac{1}{2}\|Cx-b\|^2,$$

and δ_H is the indicator function of the hyperplane $H = \{x \in \mathbf{R}^n \mid \mathbf{1}^T x = 1\}$. We use the relative entropy distance

$$d_{\mathrm{p}}(x,y) = \sum_{i=1}^n (x_i \log(x_i/y_i) - x_i + y_i), \qquad \mathbf{dom}\, d_{\mathrm{p}} = \mathbf{R}^n_+ \times \mathbf{R}^n_{++}.$$

in the primal space. This distance is 1-strongly convex with respect to ℓ_1-norm [BT09] (and also ℓ_2-norm). With the relative entropy distance, all the primal iterates $x^{(k)}$ remain feasible. In the dual space we use the Euclidean distance. Thus, the matrix norm (31) in the stepsize condition (38) for the Bregman Condat–Vũ algorithms is the (1,2)-operator norm

$$\|A\|_{1,2} = \sup_{v\neq 0} \frac{\|Av\|}{\|v\|_1} = \max_{i=1,\ldots,n} \|a_i\| = \sqrt{2},$$

where a_i is the ith column of A. In the Bregman PD3O algorithm, we use the squared Euclidean distance $d_{\mathrm{p}}(x,y) = \frac{1}{2}\|x-y\|^2$, and the matrix norm in the stepsize condition (71) is the the spectral norm $\|A\|_2$. It is easily shown that for the difference matrix (80), $\|A\|_2$ is bounded above by 2, and very close to this upper bound for large n.

The Lipschitz constant for h with respect to the ℓ_1-norm is the largest absolute value of the elements in C^TC, *i.e.*, $L_1 = \max_{i,j} |(C^TC)_{ij}|$. This value is used in the stepsize condition (38) for the Bregman Condat–Vũ algorithms. The Lipschitz constant with respect to the ℓ_2-norm is $L_2 = \|C\|_2$, which is used in the stepsize condition (71) for Bregman PD3O.

The matrix norms and Lipschitz constants are summarized as follows:

	matrix norm	Lipschitz constant
Bregman Condat–Vũ	$\|A\|_{1,2} = \sqrt{2}$	$L_1 = \max_{i,j} \|(C^TC)_{ij}\|$
Bregman PD3O	$\|A\|_2 \le 2$	$L_2 = \|C\|_2$.

The two matrix norms are either known or easy to estimate, and we directly calculate L_1 and L_2 in this example. But in general the Lipschitz constant might be expensive to compute or estimate, *e.g.*, via the power method.

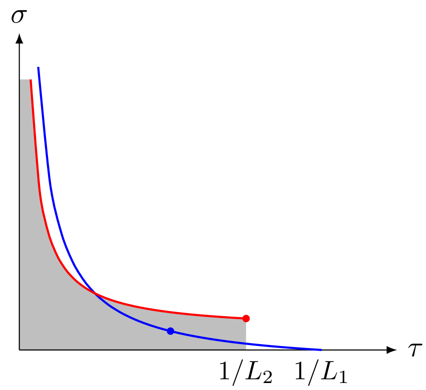

Figure 9: Illustration of ranges of parameters in Bregman Condat–Vũ and Bregman PD3O for the problem (79). The gray region under the red curve shows the range of possible parameters for Bregman PD3O, while the region under the blue curve shows that for Bregman Condat–Vũ. In this example, the two regions are not comparable due to different definitions of the matrix norm and the Lipschitz constant. The blue and red points indicate the chosen parameters in (82), which satisfy the requirement (68) with equality.

The Bregman proximal operator of f has a closed-form solution:

$$\operatorname{prox}_f^{\phi}(y, a) = \frac{1}{\sum_{i=1}^n y_i e^{-a_i}} \begin{bmatrix} y_1 e^{-a_1} \\ \vdots \\ y_n e^{-a_n} \end{bmatrix}, \tag{81}$$

and the (Euclidean) proximal operator of g^* is the projection onto the infinity norm ball:

$$\operatorname{prox}_{g^*}(z)_i = \begin{cases} \lambda & z_i > \lambda \\ z_i & |z_i| \leq \lambda \\ -\lambda & z_i < -\lambda. \end{cases}$$

The experiment is carried out in Python 3.6 on a desktop with an Intel Core i5 2.4GHz CPU and 8GB RAM. We set $m = 500$ and $n = 10,000$. The elements in the matrix $C \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$ are randomly generated from independent standard Gaussian distribution. For the constant stepsize option, we choose

$$\begin{array}{lll} \text{Condat-Vũ} & \sigma = L_1/2 & \tau = 1/(2L_1) \\ \text{PD3O} & \sigma = L_2/4 & \tau = 1/L_2. \end{array} \tag{82}$$

These two choices, as well as the range of possible parameters, are illustrated in Figure 9. These two choices are on the blue and red curve, respectively, and satisfy the requirement (68) with equality. For the line search algorithm, we set $\bar{\theta}_k = 1.2$ to encourage more aggressive updates, and $\beta = \sigma_{-1}/\tau_{-1} = L_1^2$, which is consistent with the choice in (82).

Numerical results We solve the problem (79) using the Bregman primal Condat–Vũ algorithm (27), the Bregman dual Condat–Vũ algorithm with line search (55), and Bregman PD3O (18).

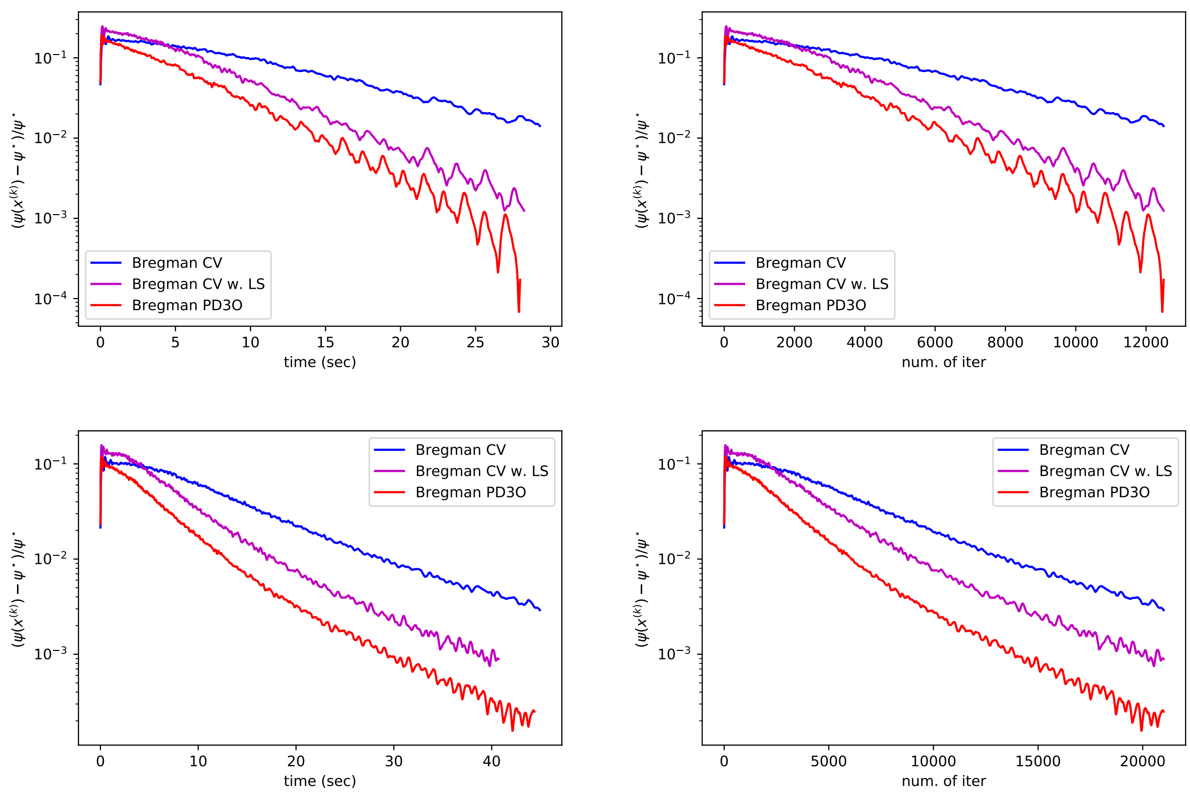

Figure 10: Comparison of three algorithms (Bregman primal Condat–Vũ, Bregman dual Condat–Vũ with line search, and Bregman PD3O) in terms of objective values. The top two figures plot the relative error of the function value versus CPU time and number of iterations for one problem instance (79), respectively. The bottom two figures correspond to another problem instance.

Figure 10 reports the relative distance between the function values to the optimal value $\psi^\star$, which is computed via CVXPY [DCB14]. Comparison between the Bregman primal Condat–Vũ algorithm and Bregman PD3O shows that Bregman PD3O converges faster. Figure 10 also compares the performance between the Bregman primal Condat–Vũ algorithm with constant stepsizes and Bregman dual algorithm with line search. One can see clearly that the line search helps convergence significantly because the more aggressive stepsizes we use is still valid locally, but may not satisfy the global requirement in (38). On the other hand, the line search does not need much computation overhead, as the plots of the CPU time and the number of iterations are roughly identical. We also observe from experiments that the performance of Bregman PD3O and the Bregman dual Condat–Vũ algorithm with line search is similar: one may outperform the other in some instances while converges much more slowly when the data changes.

8 Conclusions

We presented two variants of Bregman Condat–Vũ algorithms, introduced a line search technique for the Bregman dual Condat–Vũ algorithm, and proposed a Bregman extension to PD3O. These new methods offer the possibility to match the Bregman distance to the structure in a wider range of optimization problems. Suitable designs of Bregman distances and interesting applications of the proposed algorithms to various real-world problems would be a promising direction of research.

There are still many other open questions. For example, the connection between Bregman PDHG and Bregman Douglas–Rachford algorithm is still unknown. Also, it is still unclear how to embed Bregman distances in PDDY, and how to apply the line search technique to Bregman PD3O and to solving the general problem (1).

References

[ADH+21] D. Applegate, M. Dóaz, O. Hinder, H. Lu, M. Lubin, B. O'Donoghue, and W. Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *arXiv*, 2021.

[AT06] A. Auslender and M. Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006.

[BBT17] H. H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

[Bré73] H. Brézis. *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*, volume 5 of *North-Holland Mathematical Studies*. North-Holland, 1973.

[BSTV18] J. Bolte, S. Sabach, S. Teboulle, and Y. Vaisbourd. First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM in Optimization*, 2018.

[BT09] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery. In Y. Eldar and D. Palomar, editors, *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2009.

[Bub15] S. Bubeck. Convex optimization: algorithms and complexity. *Foundations and Trends Machine Learning*, 8(3–4):231–357, November 2015.

[CHZ13] P. Chen, J. Huang, and X. Zhang. A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems*, 29(2), 2013.

[CLMW19] C. Clason, D. A. Lorenz, Hinrich Mahler, and B. Wirth. Entropic regularization of continuous optimal transport problems. *arXiv e-prints*, June 2019.

[Con13] 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.

[CP07] 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.

[CP11a] 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.

[CP11b] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pages 185–212. Springer New York, 2011.

[CP16a] A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, pages 161–319, 2016.

[CP16b] 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.

[CT93] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3:538–543, 1993.

[CV18] H.-H. Chao and L. Vandenberghe. Entropic proximal operators for nonnegative trigonometric polynomials. *IEEE Transactions on Signal Processing*, 66(18):4826–4838, 2018.

[CZ97] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 1997.

[DCB14] S. Diamond, E. Chu, and S. Boyd. CVXPY: a Python-embedded modeling language for convex optimization, version 0.2. `cvxpy.org`, 2014.

[DST15] Y. Drori, S. Sabach, and M. Teboulle. A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. *Operations Research Letters*, 43(2):209–214, March 2015.

[DY15] D. Davis and W. Yin. A three-operator splitting scheme and its optimization applications. 2015. `arxiv.org/abs/1504.01032`.

[EB92] 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.

[Eck93] J. Eckstein. Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming. *Mathematics of Operations Research*, 18(1):202–226, 1993.

[EZC10] 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.

[Gül94] O. Güler. Ergodic convergence in proximal point algorithms with Bregman functions. In D.-Z. Du and J. Sun, editors, *Advances in Optimization and Approximation*, pages 155–165. Springer, 1994.

[HY12] 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.

[JLLO19] M. Jacobs, F. Leger, W. Li, and S. Osher. Solving large-scale optimization problems with a convergence rate independent of grid size. *SIAM Journal on Numerical Analysis*, 57(3):1100–1123, 2019.

[JV21] X. Jiang and L. Vandenberghe. Bregman primal–dual first-order method and applications to sparse semidefinite programming. *Computational Optimization and Applications*, 2021.

[KP15] N. Komodakis and J. Pesquet. Playing with duality: an overview of recent primal–dual approaches for solving large-scale optimization problems. *IEEE Signal Processing Magazine*, 32(6):31–54, November 2015.

[LM79] 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.

[LV11] I. Loris and C. Verhoeven. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 27(12), 2011.

[LXY21] Y. Liu, Y. Xu, and W. Yin. Acceleration of primal–dual methods by preconditioning and simple subproblem procedures. *Journal of Scientific Computing*, 86(2):21, 2021.

[Mor65] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Math. Soc. France*, 93:273–299, 1965.

[MP18] Y. Malitsky and T. Pock. A first-order primal-dual algorithm with linesearch. *SIAM Journal on Optimization*, 28(1):411–432, 2018.

[Nes18] Yurii Nesterov. *Lectures on Convex Optimization*. Springer, second edition, 2018.

[OHG12] H. Ouyang, N. He, and A. Gray. Stochastic ADMM for nonsmooth optimization. *ArXiv e-prints*, November 2012.

[OV18] D. O'Connor and L. Vandenberghe. On the equivalence of the primal-dual hybrid gradient method and douglas–rachford splitting. *Mathematical Programming*, 2018.

[PB13] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[PC11] 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.

[PCBC09] 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.

[SCMR20] A. Salim, L. Condat, K. Mishchenko, and P. Richtárik. Dualize, split, randomize: Fast nonsmooth optimization algorithms. *arXiv preprint*, 2020.

[ST14] 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.

[Teb18] M. Teboulle. A simplified view of first order methods for optimization. *Mathematical Programming*, 170(1):67–96, July 2018.

[Tse00] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM Journal on Control and Optimization*, 38(2):431–446, 2000.

[Tse08] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. 2008.

[VMC21] M.-L. Vladarean, Y. Malitsky, and V. Cevher. A first-order primal-dual method with adaptivity to local smoothness. In *Advances in Neural Information Processing Systems*, 2021.

[Vũ13] B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38:667–681, 2013.

[Yan18] M. Yan. A new primal-dual algorithm for minimizing the sum of three functions with a linear operator. *Journal of Scientific Computing*, 2018.

[YHA21] E. Yazdandoost Hamedani and N. S. Aybat. A primal–dual algorithm with line search for general convex-concave saddle point problems. *SIAM Journal on Optimization*, 31(2):1299–1329, 2021.