J Sci Comput (2011) 46: 20–46
DOI 10.1007/s10915-010-9408-8

A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration

Xiaoqun Zhang · Martin Burger · Stanley Osher

Received: 23 November 2009 / Revised: 7 April 2010 / Accepted: 8 April 2010 /
Published online: 5 August 2010

Abstract In this paper, we propose a unified primal-dual algorithm framework for two classes of problems that arise from various signal and image processing applications. We also show the connections to existing methods, in particular Bregman iteration (Osher et al., Multiscale Model. Simul. 4(2):460–489, 2005) based methods, such as linearized Bregman (Osher et al., Commun. Math. Sci. 8(1):93–111, 2010; Cai et al., SIAM J. Imag. Sci. 2(1):226–252, 2009, CAM Report 09-28, UCLA, March 2009; Yin, CAAM Report, Rice University, 2009) and split Bregman (Goldstein and Osher, SIAM J. Imag. Sci., 2, 2009). The convergence of the general algorithm framework is proved under mild assumptions. The applications to ℓ^1 basis pursuit, TV$-L^2$ minimization and matrix completion are demonstrated. Finally, the numerical examples show the algorithms proposed are easy to implement, efficient, stable and flexible enough to cover a wide variety of applications.

Keywords Saddle point · Bregman iteration · ℓ^1 minimization · Inexact Uzawa methods · Proximal point iteration

1 Introduction

The main goal of this paper is to propose a unified algorithm framework for two classes of convex optimization problems arising from sparse reconstruction. The framework proposed here is a continued work started in [51], where a Bregmanized operator splitting (BOS)

X. Zhang (✉)
Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, P.R. China
e-mail: xiaoqun.zhang@gmail.com

M. Burger
Institute for Computational and Applied Mathematics, Westfälische Wilhelms-Universität, Einsteinstr. 62, 48163 Münster, Germany
e-mail: martin.burger@wwu.de

S. Osher
Department of Mathematics, UCLA, Box 951555, Los Angeles, CA 90095-1555, USA
e-mail: sjo@math.ucla.edu

method is proposed for nonlocal total variation regularization. In addition to unifying some existing algorithms, we also propose new ones such as an extension of split Bregman [32] that linearizes quadratic penalties to yield simpler iterations.

This work was originated from Bregman iteration [41], but we can find the connections to other classical optimization concepts, such as augmented Lagrangian method [44] and proximal point minimization. Bregman iteration for image processing problems was originally proposed by Osher, Burger, Goldfarb, Xu and Yin in [41] to improve the classical Rudin-Osher-Fatemi [45] total variation (TV) regularization model for image restoration. For a given closed, proper convex functional $J(u):\mathbb{R}^N \to R\cup\{+\infty\}$, the Bregman distance [7] is defined as

$$D_J^p(u,v) = J(u) - J(v) - \langle p, u-v\rangle, \tag{1.1}$$

where $p \in \partial J(v)$ is some subgradient of J at the point v and $\langle\cdot,\cdot\rangle$ denotes the canonical inner product in $\mathbb{R}^N$. It is well known that Bregman distance (1.1) is not a distance in the usual sense since it is generally not symmetric. However, it measures the closeness of two points since $D_J^p(u,v) \ge 0$ for any u and v. Furthermore, if the functional J is strictly convex, the following relation is satisfied:

$$D_J^p(u,v) = 0 \quad \text{iff } u = v.$$

We consider a standard unconstrained problem

$$\min_{x\in\mathbb{R}^N}\left(\mu J(x) + \frac{1}{2}\|Ax-b\|^2\right), \tag{1.2}$$

where $A \in R^{M\times N}$, μ is a positive number, x is the unknown signal/image, b is typically an input noisy measurement, and $\|\cdot\|$ denotes the ℓ^2 norm in $\mathbb{R}^M$. We are also interested in the related equality constrained problem

$$\min_{x\in\mathbb{R}^N} J(x) \quad \text{s.t. } Ax = b. \tag{1.3}$$

Using the Bregman distance (1.1), an iterative regularization method is proved in [41] to solve (1.3):

$$\begin{cases} x^{k+1} = \min_x(\mu D_J^{p^k}(x,x^k) + \frac{1}{2}\|Ax-b\|^2) \\ p^{k+1} = p^k + \frac{1}{\mu}A^\top(b - Ax^{k+1}), \end{cases} \tag{1.4}$$

where $p^{k+1} \in \partial J(x^{k+1})$ and $A^\top$ is the adjoint operator of A. By a change of variables under certain assumptions, the above algorithm can be simplified as

$$\begin{cases} x^{k+1} = \min_x(\mu J(x) + \frac{1}{2}\|Ax-y^k\|^2) \\ y^{k+1} = y^k + b - Ax^{k+1}, \end{cases} \tag{1.5}$$

for $k = 0, 1, \dots$ starting with $x^0 = 0$, $y^0 = b$. From (1.5), the constrained problem (1.3) can be solved by a sequence of unconstrained subproblems such as (1.2) and gradient ascent steps. There are two main convergence results for the sequence $\{x^k\}$ generated by (1.4):

$$\|Ax^k - b\| \searrow 0 \quad \text{and} \quad D_J^{p^k}(\overline{x}, x^k) \to 0,$$

where $\overline{x}$ is a true solution of the problem (1.3). In practice, when there is noise, this algorithm still can be applied with a stopping criterion according to a discrepancy principle of the residual $\|Ax^k - b\|^2$. In [41], the algorithm (1.5) was shown to be efficient for improving denoising quality over the original model (1.2) when $J(u)$ is the total variation functional, since contrasts and micro-structures are refined by adding the residuals back.

Solving the first subproblem of (1.5), which is equivalent to (1.2), can sometime be difficult, since it involves inverting the operator A. There are ways to decouple the variables related by A and simplify the unconstrained subproblems involving J. The minimization of the sum of two convex functionals is still an active research area in the optimization community. In particular, by considering splitting and decomposition principles, Combettes and Wajs in [20] applied a *proximal forward-backward splitting* (PFBS) technique based on the proximal operator introduced by Moreau in [40]. The PFBS algorithm [20] applied to (1.2) is described as

$$x^{k+1} = \mathrm{Prox}_{\delta\mu J}(x^k - \delta A^\top(Ax^k - b)), \tag{1.6}$$

for a positive number $0 < \delta < \frac{2}{\|A^\top A\|}$ where the proximal operator is defined by

$$\mathrm{Prox}_J(v) := \arg\min_u \left(J(u) + \frac{1}{2}\|u - v\|^2 \right). \tag{1.7}$$

A main advantage of the proximal operator is that the subproblem (1.7) becomes strictly convex. Then there exists a unique minimizer $\hat{x} = \mathrm{Prox}_J(v)$ for any v. By introducing an auxiliary variable z^k, the solution of the minimization problem (1.2) can be computed by the following two-step algorithm:

$$\begin{cases} z^{k+1} = x^k - \delta A^\top(Ax^k - b), \\ x^{k+1} = \mathrm{Prox}_{\delta\mu J}(z^{k+1}). \end{cases} \tag{1.8}$$

The proximal operator solution (1.7) has well known solutions for some models. For example, when the regularization functional J is the ℓ^1 norm of x, i.e. $J(x) = \|x\|_1$, then the solution for the second subproblem is obtained by a soft shrinkage operator [20, 22, 34] as follows:

$$x^{k+1} = \mathrm{shrinkage}(z^{k+1}, \delta\mu) := \mathrm{sign}(z^{k+1}) \max\{|z^{k+1}| - \delta\mu, 0\} \tag{1.9}$$

where all the operations are pointwise performed. This idea of using operator splitting and Bregman iteration to benefit from the efficiency of soft thresholding is thus applied to the ℓ^1 basis pursuit problem in [11, 42, 50]. The overall algorithm PFBS + Bregman for solving (1.3) is then

$$\begin{cases} \begin{cases} z^{k+1} = x^k - \delta A^\top(Ax^k - y^k), \\ x^{k+1} = \mathrm{Prox}_{\delta\mu}(z^{k+1}), \end{cases} \\ y^{k+1} = y^k + (b - Ax^{k+1}). \end{cases} \tag{1.10}$$

From (1.10), we can see that in general we need an infinite number of inner iterations to get full convergence of x^{k+1} before we update y^{k+1}. The linearized Bregman in [9, 11, 21, 42] and the Bregmanized Operator Splitting (BOS) in [51] are actually obtained by applying only one inner step for the subproblem (1.2). In particular, linearized Bregman is formulated as

$$\begin{cases} x^{k+1} = \mathrm{Prox}_{\delta\mu J}(\delta A^\top y^k), \\ y^{k+1} = y^k + (b - Ax^{k+1}), \end{cases} \tag{1.11}$$

and the BOS algorithm is

$$\begin{cases} z^{k+1} = x^k - \delta A^\top (Ax^k - y^k), \\ x^{k+1} = \mathrm{Prox}_{\delta\mu J}(z^{k+1}), \\ y^{k+1} = y^k + (b - Ax^{k+1}). \end{cases} \tag{1.12}$$

It is proved that BOS (1.12) converges to the solutions of (1.3), while linearized Bregman (1.11) is proved to converge to the solutions of an approximated strictly convex functional

$$\tilde{J}(x) = \mu J(x) + \frac{1}{2\delta}\|x\|^2 \quad \text{s.t. } Ax = b. \tag{1.13}$$

When $J(x) = \|x\|_1$, it was proved in [49] that, for μ sufficient large, linearized Bregman solves the original problem (1.3). One goal of this paper is to generalize BOS algorithm in a primal-dual framework and establish the connections to some existing methods such as augmented Lagrangian methods [44] and proximal point minimization. Also, we will investigate the applications of BOS algorithms in ℓ^1 basis pursuit in Sect. 5.1 and matrix completion problem in Sect. 6.1.

Another class of problem we are interested in is related to the split Bregman method for discrete TV regularization proposed by Goldstein and Osher in [32]. The idea is to reformulate the original TV minimization problem as ℓ^1 minimization to benefit from the efficiency of the soft thresholding operation (1.9). Consider anisotropic TV minimization problem

$$\min_{x\in\mathbb{R}^N} \|\nabla x\|_1 + \frac{\lambda}{2}\|Ax - b\|^2,$$

where $\nabla x \in \mathbb{R}^{2N}$ denotes the discrete gradient of $x \in \mathbb{R}^N$ and $\|\nabla u\|$ denotes the sum of the length of gradient vector at each point. We can rewrite the above problem as a separable equality constrained problem

$$\min_{x\in\mathbb{R}^N, z\in\mathbb{R}^{2N}} \|z\|_1 + \frac{\lambda}{2}\|Ax - b\|^2, \quad \text{s.t. } \nabla x = z. \tag{1.14}$$

By using Bregman iteration to handle the introduced constraint and an alternating approach to approximate the minimization over x, z, the split Bregman algorithm in [32] is given as

$$\begin{cases} x^{k+1} = \arg\min_x (\frac{\lambda}{2}\|Ax - b\|^2 + \frac{\delta}{2}\|\nabla x - z^k + y^k\|^2), \\ z^{k+1} = \arg\min_z (\|z\|_1 + \frac{\delta}{2}\|z - \nabla x^{k+1} - y^k\|^2), \\ y^{k+1} = y^k + (\nabla x^{k+1} - z^{k+1}). \end{cases} \tag{1.15}$$

Since the first subproblem only involves a least square type problem, it can sometimes be efficiently solved, especially when A is a diagonalizable operator, such as a Fourier sub-sampling matrix used in compressive sensing, or a convolution operator with periodic boundary conditions. The second subproblem is the ℓ^1 norm minimization. Thus it can be solved by the soft thresholding operator (1.9). Furthermore, the algorithm is easy to implement compared to other Newton based method such as [16]. The split Bregman method is also applied to segmentation and surface reconstruction in [31]. A similar splitting approach is applied for TV$-L^1$ minimization in [33, 48]. On the other hand, the convergence of the split Bregman is not justified in [32]. Recently, several authors discussed the convergence of this method. In [26, 46], the connection between split Bregman and Douglas-Rachford

Spitting [24], is pointed out, clarifying the convergence. In [10], the authors justified the convergence of the alternating algorithm.

Interestingly, there is an extensive literature from the optimization community about methods for minimizing convex functionals with linear constraints such as (1.3) and problems with separable structures such as (1.14). For example, the theoretical connection between Bregman iteration (1.5) and the classical augmented Lagrangian method is discussed in [50]. Such multiplier methods and proximal point algorithms are largely studied in classical nonlinear programming literature, such as by Arrow, Hurwicz and Uzawa in [1], by Rockafellar [44], by Glowinski and Le Tallec in [30] and references therein.

In this paper we will introduce a unified primal dual algorithm framework for two classes of problems taking the form (2.5) and (2.6). The framework is based on the generalized proximal point algorithm and the augmented Lagrangian method, which we call it as a variant of inexact Uzawa method [1, 6]. There are several advantages of this framework. First, most convergence results of classical Uzawa methods and Bregman iteration methods are based on assumptions of the strict convexity of the original functionals, while the convergence of our proposed algorithms are guarantied without requiring strict convexity. Secondly, the proposed algorithms can be applied in different settings when computing the inverse of a linear operator becomes inefficient, thus it is suitable for a very large class of problems. Finally, compared to the popular linearized Bregman method, the proposed algorithm solves the original problem without approximation, as a result it is robust to the choice of the parameter. Notice that we are not declaring that we will propose a faster algorithm than existing methods, but we are proposing a simple and flexible algorithm framework for different applications under minimum conditions. Also, some proposed algorithms in the framework are apparently new.

The paper is organized as follows. First, we introduce the primal dual formulations of the two classes of problems and briefly review some existing methods. Then we present two generic algorithms under a unified framework and discuss their convergence. Finally, we present several applications in signal/image processing, such as ℓ^1 basis pursuit, TV-L^2 minimization and extension to matrix completion. Furthermore, we compare the proposed algorithms with existing popular Bregman based methods, such as linearized Bregman [9] for ℓ^1 basis pursuit, SVT for matrix completion [8] and split Bregman [32] for TV regularization.

2 Primal-Dual Formulation

Before we consider the specific problems, we first introduce a general saddle point problem notation. We consider a convex-concave function $L(x; y) : X \times Y \to \mathbb{R}$, where X and Y are closed convex sets in $\mathbb{R}^N$ and $\mathbb{R}^M$. We are interested in computing a saddle point pair $(\overline{x}, \overline{y})$ that satisfies

$$L(\overline{x};\ y) \leq L(\overline{x}; \overline{y}) \leq L(x; \overline{y}), \quad \forall x \in X,\ y \in Y. \tag{2.1}$$

The primal convex problem is denoted by

$$\inf_{x \in X} J(x) \tag{2.2}$$

where $J(x) = \sup_{y \in Y} L(x; y)$, and the dual problem is

$$\sup_{y \in Y} q(y) \tag{2.3}$$

where $q(y) = \inf_{x\in X} L(x; y)$. The pair $(\overline{x};\ \overline{y})$ is a saddle point of the Lagrangian function $L(x;\ y)$ if and only if $\overline{x}$ is an optimal solution of the primal problem (2.2) and $\overline{y}$ is a solution of the dual problem (2.3) respectively. In general, the existence of solutions for a primal problem does not imply the existence of saddle points of Lagrangian functions, but the converse holds, see [30]. Here, we assume the existence of solutions to the original primal problems and related saddle point problems.

Classical methods based on the gradient/subgradient for solving the above saddle point problems have been of great interest since the seminal work of Arrow, Hurwicz and Uzawa [1]. These methods alternate a minimization of the Lagrangian function $L(x;\ y)$ with respect to x given y and gradient ascent with respect to y given x. More precisely, a general Uzawa method can be written as

$$\begin{cases} x^{k+1} = \arg\min_{x\in X} L(x, y^k), \\ y^{k+1} = P_Y(y^k + \gamma\partial_y L(x^{k+1};\ y^k)) \end{cases} \tag{2.4}$$

where $\partial_y L(x^{k+1};\ y^k)$ is a subgradient of $L(x;\ y)$ with respect to y at $(x^{k+1};\ y^k)$, $P_Y(\cdot)$ is the orthogonal projection operator to Y and γ is a positive number. The convergence is established assuming strict convexity. This method and its variants are largely studied for quadratic programming in the literature due to the simplicity. This method is in fact a gradient-type algorithm and under certain conditions, a linear convergence rate can be achieved [30]. Particularly, a class of inexact Uzawa methods are proposed and analyzed in [6] for linear saddle points problems by introducing preconditioning matrices in order to achieve fast convergence. Our proposed methods are also based on the idea of the general Uzawa method and we attempt to achieve the convergence without strict convexity assumptions.

Now, we consider two general convex programming problems which cover various signal and image processing applications:

$$(P_0) \qquad \min_{x\in\mathbb{R}^N} J_0(x) \quad \text{s.t. } Ax = b \tag{2.5}$$

as in (1.3) and

$$(P_1) \qquad \min_{x\in\mathbb{R}^N, z\in\mathbb{R}^K} \Big(J_1(z) + H(x)\Big) \quad \text{s.t. } Bx = z \tag{2.6}$$

as (1.14), where $J_0 : \mathbb{R}^N \to (-\infty, \infty]$, $J_1 : \mathbb{R}^K \to (-\infty, \infty]$ and $H : \mathbb{R}^N \to (-\infty, \infty]$ are closed proper convex functionals, and $A \in \mathbb{R}^{M\times N}$, $B \in \mathbb{R}^{K\times N}$.

The Lagrangian function for problem (2.5) and problem (2.6) are respectively

$$L_0(x;\ y) = J_0(x) - \langle y, Ax - b\rangle \tag{2.7}$$

and

$$L_1(x, z;\ y) = J_1(z) + H(x) - \langle y, z - Bx\rangle, \tag{2.8}$$

where y is the Lagrange multiplier associated with the respective constraint in each problem. Note that the multiplier variables in the two problems are in different sets $\mathbb{R}^M$ and $\mathbb{R}^K$, but here we use the same notation y without confusion. The related dual problems are respectively defined as

$$(D_0) \qquad \min_y J_0^*(A^\top y) - \langle b, y\rangle, \tag{2.9}$$

$$(D_1) \quad \min_y J_1^*(y) + H^*(-B^\top y), \tag{2.10}$$

where J_0^*, J_1^* and H^* denote the Legendre-Fenchel transforms of the convex functional J_0, J_1 and H respectively, defined by

$$J_0^*(A^\top y) = \sup_x \langle A^\top y,\ x\rangle - J_0(x),$$

$$J_1^*(y) = \sup_z \langle y,\ z\rangle - J_1(z),$$

$$H^*(-B^\top y) = \sup_x \langle -B^\top y,\ x\rangle - H(x).$$

In the following, we will focus on the Lagrangian formulations and briefly review some primal-dual methods related to ours. Among them, the augmented Lagrangian method, also called *method of multipliers* [35, 43, 44], plays an important role. By adding a quadratic penalty to the Lagrangian formulation (2.7), we obtain the augmented Lagrangian formulation for problem (2.5)

$$L_0^\alpha(x, y) = J_0(x) - \langle y, Ax - b\rangle + \frac{\alpha}{2}\|Ax - b\|^2. \tag{2.11}$$

Similarly modifying the Lagrangian from (2.8), the augmented Lagrangian function for problem (2.6) is written as

$$L_1^\alpha(x, z, y) = J_1(z) + H(x) - \langle y, z - Bx\rangle + \frac{\alpha}{2}\|Bx - z\|^2. \tag{2.12}$$

We can observe that any saddle point of $L_0(x;\ y)$ and $L_1(x, z;\ y)$ are saddle points of $L_0^\alpha(x;\ y)$ and $L_1^\alpha(x, z;\ y)$ respectively and that the converse also holds. The *method of multipliers* [35, 43] is then obtained by applying the general Uzawa algorithm (2.4) on the augmented Lagrangian formulation and setting $\gamma = \alpha$. More precisely, for (2.5) we get the sequence (x^k, y^k)

$$\begin{cases} x^{k+1} = \arg\min_x (J_0(x) - \langle y^k, Ax\rangle + \frac{\alpha}{2}\|Ax - b\|^2), \\ y^{k+1} = y^k + \alpha(b - Ax^{k+1}) \end{cases} \tag{2.13}$$

and for (2.6) we get the sequence (x^k, z^k, y^k)

$$\begin{cases} (x^{k+1}, z^{k+1}) = \arg\min_{x,z} (J_1(z) + H(x) - \langle y^k, z - Bx\rangle + \frac{\alpha}{2}\|Bx - z\|^2), \\ y^{k+1} = y^k + \alpha(Bx^{k+1} - z^{k+1}). \end{cases} \tag{2.14}$$

Using the augmented Lagrangian function, faster convergence can be achieved compared to Lagrangian functions by carefully choosing the parameters. For example, the method achieves a superlinear convergence when $\alpha \nearrow \infty$ [30], although this may cause instability due to ill conditioning. One advantage of this method is that the convergence does not require a strict convexity of the functionals. It has been shown to be equivalent to a proximal point algorithm applied to the dual [44]. However, the algorithm (2.14) could be difficult to solve since it involves the minimization over x and z simultaneously because of the quadratic term $\|Bx - z\|^2$. To take advantage of the separable structure of problem (P_1), we can apply the *alternating direction method of multipliers* (ADMM) introduced by Gabay in [29]. This approximates the (x,z) minimization (2.14) with minimization over x with z fixed and z with x fixed. That is

$$\begin{cases} x^{k+1} = \arg\min_x (H(x) + \langle y^k, Bx\rangle + \frac{\alpha}{2}\|Bx - z^k\|^2), \\ z^{k+1} = \arg\min_z (J_1(z) - \langle y^k, z\rangle + \frac{\alpha}{2}\|Bx^{k+1} - z\|^2), \\ y^{k+1} = y^k + \alpha(Bx^{k+1} - z^{k+1}). \end{cases} \tag{2.15}$$

This method can be interpreted from different perspectives. For example, it is equivalent to Douglas-Rachford splitting method [24]. See [25, 30] and references therein for more variants, such as exchanging the roles of the variables x and z, and adding a multiplier update between x and z to get a more symmetrized method. In particular, when $J_1(z) = \|z\|_1$, $H(x) = \frac{\lambda}{2}\|Ax - b\|^2$ and $B = \nabla$ as in problem (1.14), the split Bregman method (1.15) proposed in [32] is equivalent to the algorithm (2.15).

However, both the augmented method (2.13) for (P_0) and the alternating method (2.15) for Problem (P_1) involve inverting of A or B, which is usually not practical for getting an accurate solution at a reasonable cost. This requires us to develop a decoupled algorithm which does not involve the inverse of A and B. A related algorithm called the *predictor corrector proximal multiplier method* (PCPM) developed by Chen and Teboulle [18] is obtained by introducing an auxiliary variable and iterating

$$\begin{cases} p^{k+1} = y^k + \alpha(Bx^k - z^k), \\ x^{k+1} = \arg\min_x (H(x) + \langle p^{k+1}, Bx\rangle + \frac{1}{2\delta}\|x - x^k\|^2), \\ z^{k+1} = \arg\min_z (J_1(z) - \langle p^{k+1}, z\rangle + \frac{1}{2\delta}\|z - z^k\|^2), \\ y^{k+1} = y^k + \alpha(Bx^{k+1} - z^{k+1}). \end{cases} \tag{2.16}$$

Since the minimization of x and z does not involve inverting the matrix B, and the steps decouple, it is thus possible to parallelize. However, this method is in general slow in practice due to the fact that the two primal variables x and z are updated independently based on previous iteration instead of most recent update.

The algorithm framework proposed in this paper combines the decoupling of PCPM and the efficiency of augmented Lagrangian methods. Our goal is to develop a general algorithm that only requires mild regularity assumptions and allows for simple stepsize choosing.

3 Proposed General Algorithms

Our algorithms are based on the principle of Bregman proximal iteration [14, 17, 41] and Moreau-Yosida regularization [37]. The idea is to generalize the proximal minimization (1.7) by replacing the quadratic penalty with the Bregman distance of a strictly convex and continuously differentiable function ϕ. The proximal minimization of a convex function $J(x)$ becomes

$$x^{k+1} = \arg\min_x \Big(J(x) + D_\phi^{p^k}(x, x^k)\Big), \tag{3.1}$$

where $D_\phi^{p^k}(x, x^k)$ is the Bregman distance defined as in (1.1), p^k is the unique gradient of $\phi(x)$ at x^k since we assume ϕ is differentiable. In order to design effective algorithms, here we are only interested in the quadratic case. In particular, we choose a positive semi-definite matrix Q, and we define

$$\|x\|_Q := \sqrt{\langle Qx, x\rangle}.$$

Then we define $\phi(x) = \|x\|_Q^2$ and note that

$$D_\phi^{p^k}(x, x^k) = \|x - x^k\|_Q^2.$$

If Q is positive definite, then $\|\cdot\|_Q$ is a norm.

The general idea of our algorithms is to replace the augmented Lagrangian primal minimizations (2.13) and (2.15) by proximal-like iterations. More precisely, we propose the following two algorithms for (P_0) and (P_1):

Algorithm (A_0) for Problem (P_0)

$$x^{k+1} = \arg\min_x \left(J_0(x) - \langle y^k, Ax\rangle + \frac{\alpha}{2}\|Ax - b\|^2 + \frac{1}{2}\|x - x^k\|_{Q_0}^2\right) \tag{3.2a}$$

$$Cy^{k+1} = Cy^k + (b - Ax^{k+1}) \tag{3.2b}$$

Algorithm (A_1) for Problem (P_1)

$$x^{k+1} = \arg\min_x \left(H(x) + \langle y^k, Bx\rangle + \frac{\alpha}{2}\|Bx - z^k\|^2 + \frac{1}{2}\|x - x^k\|_{Q_1}^2\right) \tag{3.3a}$$

$$z^{k+1} = \arg\min_z \left(J_1(z) - \langle y^k, z\rangle + \frac{\alpha}{2}\|Bx^{k+1} - z\|^2 + \frac{1}{2}\|z - z^k\|_{Q_2}^2\right) \tag{3.3b}$$

$$Cy^{k+1} = Cy^k + (Bx^{k+1} - z^{k+1}) \tag{3.3c}$$

where Q_i, for $i = 0, 1, 2$ and C are positive definite matrices.

We then have the freedom to choose different matrices to cancel out or precondition the operators A and B, which might be difficult to invert or ill conditioned in real applications. Another advantage of choosing such proximal minimizations is to approximate the original objective functionals by strictly convex ones when the original functions $J_0(\cdot)$, $J_1(\cdot)$ and $H(\cdot)$ are not strictly convex. By introducing the strictly convex proximal term, the minimizer of each subproblem is unique.

The algorithms proposed can be classified as inexact Uzawa methods [6]. An example of the algorithm (A_0) is BOS proposed in [51] when we choose $Q_0 = \frac{1}{\delta} - \alpha A^\top A$. The matrix C is a precondition matrix on the dual variable y, but here for simplicity we choose simple matrices such as constant matrix. In the following, we discuss some connections with other existing methods.

- For (A_0), if we allow $Q_0 = 0$, and $C = \frac{1}{\alpha}$, the algorithm is the method of multipliers (2.13)

$$\begin{cases} x^{k+1} = \arg\min_x (J(x) + \frac{\alpha}{2}\|Ax - (b + \frac{1}{\alpha}y^k)\|^2), \\ y^{k+1} = y^k + \alpha(b - Ax^{k+1}). \end{cases} \tag{3.4}$$

  It is equivalent to the Bregman iteration method (1.5) with a different initialization. The equivalence of the augmented Lagrangian method and Bregman iteration has been pointed out by several authors [26, 47].
- When $Q_i = Id$, $C = \frac{1}{\gamma}$, the algorithms are classical proximal point algorithms on the augmented Lagrangian formulation proposed by Rockafellar in [44]. This method can overcome non-strict convexity of each functional and results in a more stable algorithm.

However, as the method of multipliers, the iterations for x involve inverse of the operator A and B for both problems.

- For (A_1), if we choose $Q_1 = \frac{1}{\delta} - \alpha B^T B$, $Q_2 = 0$, and $C = \frac{1}{\gamma}$ then we obtain the algorithm:

$$\begin{cases} p^{k+1} = y^k + \alpha(Bx^k - z^k), \\ x^{k+1} = \arg\min_x (H(x) + \langle p^{k+1}, Bx\rangle + \frac{1}{2\delta}\|x - x^k\|^2), \\ z^{k+1} = \arg\min_z (J_1(z) - \langle y^k, z\rangle + \frac{\alpha}{2}\|z - Bx^{k+1}\|^2), \\ y^{k+1} = y^k + \gamma(Bx^{k+1} - z^{k+1}). \end{cases} \tag{3.5}$$

  Compared to the PCPM algorithm (2.16), we update z^{k+1} using Bx^{k+1} instead of z^k. This algorithm also differs from the standard proximal point method applied on the Lagrangian function due to the first "predictor" step.
- The algorithms (A_0) and (A_1) can be extended to solve other problems. For example, we can combine the two algorithms to solve the mixed problem,

$$(P_2) \quad \min_{x,z} J(z) \quad \text{s. t. } Ax = b, \;\; Bx = z \tag{3.6}$$

  by using two Lagrange multipliers. The idea can also be extended to other primal dual formulations with more complicated constraint sets. Some examples are discussed in [27] which uses a variant of the PDHG method [52] and shows it to be a special case of the algorithm (3.3) proposed here.

4 Convergence Analysis

In this section, we establish the global convergence of algorithms (A_0) and (A_1).

The proof for the algorithm (A_0) is similar to the one of BOS presented in [51], and the proof for the algorithm (A_1) is similar to [51] and [10]. For completeness, we will include short proofs for the two general algorithms.

In the following, we denote the smallest and largest eigenvalues of the matrix C as λ_m^C and λ_M^C.

Theorem 4.1 *Let the sequences (x^k, y^k) be generated by the algorithm (A_0) given in (3.2). If $0 < \frac{1}{\lambda_m^C} < 2\alpha$, Q_0 be positive definite, C be positive definite, then*

- $\lim_k \|Ax^k - b\| = 0$.
- $\lim_k J_0(x^k) = J_0(\overline{x})$.
- *All limit points of (x^k, y^k) are saddle points of $L_0(x;\; y)$ defined in (2.7).*

Proof We denote $w = (x, y)$ and define a norm on w by

$$\|w\|^2 = \|x\|_{Q_0}^2 + \|y\|_C^2.$$

Let us recall that all the saddle-points of L_0, $\overline{w} = (\overline{x}, \overline{y})$, are characterized by

$$\overline{s} - A^{\top}\overline{y} = 0, \qquad A\overline{x} - b = 0, \tag{4.1}$$

where $\overline{s} \in \partial J_0(\overline{x})$.

Let $w^k = (x^k,\ y^k)$ be a sequence generated by the algorithm (A_0). We define $s^{k+1} = A^\top y^k - \alpha A^\top (Ax^{k+1} - b) - Q_0(x^{k+1} - x^k)$, then we can see $s^{k+1} \in \partial J_0(x^{k+1})$ by the optimality condition of (3.2a). We denote the errors as follows:

$$\begin{aligned} s_e^{k+1} &:= s^{k+1} - \overline{s}, \\ w_e^{k+1} &:= w^{k+1} - \overline{w} = (x_e^{k+1}, y_e^{k+1}) = (x^{k+1} - \overline{x}, y^{k+1} - \overline{y}). \end{aligned} \tag{4.2}$$

Using the optimality condition (4.1), the sequence $(s_e^k,\ w_e^k)$ satisfies:

$$\begin{cases} s_e^{k+1} + \alpha A^\top A x_e^{k+1} + Q_0 x_e^{k+1} = Q_0 x_e^k + A^\top y_e^k, \\ C y_e^{k+1} = C y_e^k - A x_e^{k+1}. \end{cases}$$

Taking the inner product with x_e^{k+1} on both sides of the first equality, we get

$$\begin{aligned} &\frac{1}{2}(\|x_e^{k+1}\|_{Q_0}^2 + \|x^{k+1} - x^k\|_{Q_0}^2 - \|x_e^k\|_{Q_0}^2) \\ &\quad = -\langle s_e^{k+1}, x_e^{k+1}\rangle - \alpha \|A x_e^{k+1}\|^2 + \langle y_e^k, A x_e^{k+1}\rangle. \end{aligned}$$

Similarly, from the second equality, we can derive

$$\frac{1}{2}(\|y_e^{k+1}\|_C^2 - \|y_e^k\|_C^2 - \|A x_e^{k+1}\|_{C^{-1}}^2) = -\langle y_e^k, A x_e^{k+1}\rangle.$$

Adding the above two equalities together yields

$$\begin{aligned} &\frac{1}{2}\|w_e^{k+1}\|^2 + \frac{1}{2}\|x^{k+1} - x^k\|_{Q_0}^2 + \langle s_e^{k+1}, x_e^{k+1}\rangle + \alpha\|A x_e^{k+1}\|^2 \\ &\quad - \frac{1}{2}\|A x_e^{k+1}\|_{C^{-1}}^2) = \frac{1}{2}\|w_e^k\|^2. \end{aligned} \tag{4.3}$$

By the convexity of the functional $J_0(\cdot)$, we have $\langle s_e^{k+1}, x_e^{k+1}\rangle \geq 0$ since $s^{k+1} \in \partial J_0(x^{k+1})$ is monotone. Together with the condition $0 < \frac{1}{\lambda_m^C} < 2\alpha$, we derive

$$\langle s_e^{k+1}, x_e^{k+1}\rangle + \alpha\|A x_e^{k+1}\|^2 - \frac{1}{2}\|A x_e^{k+1}\|_{C^{-1}}^2 \geq 0,$$

which yields

$$\|w_e^{k+1}\|^2 \leq \|w_e^k\|^2.$$

This implies the boundness of the sequence $w^k = (x^k, y^k)$. Furthermore, the summation of (4.3) leads to

$$\begin{aligned} &\sum_{k=0}^{\infty}\langle s_e^{k+1}, x_e^{k+1}\rangle + \sum_{k=0}^{\infty}\frac{1}{2}\|x^{k+1} - x^k\|_{Q_0}^2 \\ &\quad + \sum_{k=0}^{\infty}\left(\alpha\|A x_e^{k+1}\|^2 - \frac{1}{2}\|A x_e^{k+1}\|_{C^{-1}}^2\right) \leq \frac{1}{2}\|w_e^0\|^2 < \infty. \end{aligned} \tag{4.4}$$

Since $\frac{1}{\lambda_m^C} < 2\alpha$, this implies also

$$\|x^{k+1} - x^k\|_{Q_0}^2 \to 0, \tag{4.5}$$

$$\|Ax_e^{k+1}\|^2 \to 0, \tag{4.6}$$

$$\langle s_e^{k+1}, x_e^{k+1} \rangle \to 0. \tag{4.7}$$

Then (4.6) implies

$$\|b - Ax^{k+1}\| = \|Ax_e^{k+1}\| \to 0. \tag{4.8}$$

(4.7) implies

$$J_0(x^{k+1}) \to J_0(\overline{x}).$$

By the boundness of w^k, there exists a subsequence w^{k_l} that converges to a point $w^\infty = (x^\infty, y^\infty)$. Now we will prove that the limit point w^∞ is a saddle point of L_0, which means that x^∞ is a solution of (P_0). We define $s^\infty := \lim_k s^{k_l}$, and we have $s^\infty \in \partial J_0(x^\infty)$ and

$$s^{k+1} - A^\top y^k = Q_0(x^k - x^{k+1}) - \alpha A^\top (Ax^{k+1} - b).$$

Taking the limit over the appropriate subsequences, we then have

$$s^\infty - A^\top y^\infty = 0,$$

$$Ax^\infty - b = 0.$$

Thus any limit point $w^\infty = (x^\infty, y^\infty)$ is a saddle point of L_0. □

In the following, we prove the convergence of the algorithm (A_1) given in (3.3).

Theorem 4.2 *Let the sequence* (x^k, z^k, y^k) *be defined by* (3.3). *Assume that* Q_1 *is positive definite*, Q_2 *is positive semi-definite, and* $0 < \frac{1}{\lambda_m^C} \le \alpha$,, *then*

- $\lim_k \|Bx^k - z^k\| = 0$.
- $\lim_k J_1(z^k) = J_1(\overline{z})$.
- $\lim_k H(x^k) = H(\overline{x})$.
- *All limit points of* (x^k, z^k, y^k) *are saddle points of* $L_1(x, z;\ y)$ (2.8).

Proof The proof is similar to the above one. With the same kind of notation for x_e^k, z_e^k, y_e^k and $s_e^k := s^k - \overline{s}$, $t_e^k := t^k - \overline{t}$ where $s^k \in \partial J_1(z^k)$, $\overline{s} \in \partial J_1(\overline{z})$, $t^k \in \partial H(x^k)$ and $\overline{t} \in \partial H(\overline{x})$. Then the sequence (x^k, z^k, y^k) generated by the algorithm (3.3) in terms of the error satisfies

$$\begin{cases} t_e^{k+1} + Q_1 x_e^{k+1} = Q_1 x_e^k - \alpha B^T (Bx_e^{k+1} - z_e^k + \frac{1}{\alpha} y_e^k), \\ s_e^{k+1} + Q_2 z_e^{k+1} = Q_2 z_e^k + y_e^k - \alpha (z_e^{k+1} - Bx_e^{k+1}), \\ Cy_e^{k+1} = Cy_e^k + (Bx_e^{k+1} - z_e^{k+1}). \end{cases} \tag{4.9}$$

By taking inner products of the above equalities with x_e^{k+1}, z_e^{k+1} and y_e^k, we then obtain

$$\frac{1}{2}(\|x_e^{k+1}\|_{Q_1}^2 + \|x^{k+1} - x^k\|_{Q_1}^2 - \|x_e^k\|_{Q_1}^2) + \langle t_e^{k+1}, x_e^{k+1} \rangle$$

$$= \alpha \left\langle Bx_e^{k+1}, z_e^k - \frac{1}{\alpha} y_e^k - Bx_e^{k+1} \right\rangle,$$

and

$$\frac{1}{2}(\|z_e^{k+1}\|_{Q_2}^2 + \|z^{k+1} - z^k\|_{Q_2}^2 - \|z_e^k\|_{Q_2}^2) + \langle s_e^{k+1}, z_e^{k+1}\rangle$$
$$= \langle y_e^k, z_e^{k+1}\rangle - \alpha\langle z_e^{k+1} - Bx_e^{k+1}, z_e^{k+1}\rangle,$$
$$\frac{1}{2}(\|y_e^{k+1}\|_C^2 - \|y_e^k\|_C^2 - \|Bx_e^{k+1} - z_e^{k+1}\|_{C^{-1}}^2) = -\langle z_e^{k+1} - Bx_e^{k+1}, y_e^k\rangle.$$

Summing over these three equalities, we obtain

$$\begin{aligned}
&\frac{1}{2}(\|x_e^{k+1}\|_{Q_1}^2 + \|x^{k+1} - x^k\|_{Q_1}^2 - \|x_e^k\|_{Q_1}^2) + \langle t_e^{k+1}, x_e^{k+1}\rangle \\
&\quad + \frac{1}{2}(\|z_e^{k+1}\|_{Q_2}^2 + \|z^{k+1} - z^k\|_{Q_2}^2 - \|z_e^k\|_{Q_2}^2) + \langle s_e^{k+1}, z_e^{k+1}\rangle \\
&\quad + \frac{1}{2}(\|y_e^{k+1}\|_C^2 - \|y_e^k\|_C^2) \\
&= \alpha\langle Bx_e^{k+1}, z_e^k - Bx_e^{k+1}\rangle - \alpha\langle z_e^{k+1} - Bx_e^{k+1}, z_e^{k+1}\rangle + \frac{1}{2}\|Bx_e^{k+1} - z_e^{k+1}\|_{C^{-1}}^2 \\
&= \left(\frac{1}{2}\|Bx_e^{k+1} - z_e^{k+1}\|_{C^{-1}}^2 - \frac{\alpha}{2}\|Bx_e^{k+1} - z_e^{k+1}\|^2\right) \\
&\quad - \frac{\alpha}{2}\|Bx_e^{k+1} - z_e^k\|^2 + \frac{\alpha}{2}(\|z_e^k\|^2 - \|z_e^{k+1}\|^2).
\end{aligned} \tag{4.10}$$

By the convexity of the functionals $J_1(\cdot)$ and $H(\cdot)$, we have

$$\langle t_e^{k+1}, x_e^{k+1}\rangle \geq 0, \qquad \langle s_e^{k+1}, z_e^{k+1}\rangle \geq 0$$

and the condition $0 < \frac{1}{\lambda_m^C} \leq \alpha$ yields

$$\frac{1}{2}\|Bx_e^{k+1} - z_e^{k+1}\|_{C^{-1}}^2 - \frac{\alpha}{2}\|Bx_e^{k+1} - z_e^{k+1}\|^2 \leq 0.$$

The equality (4.10) then implies

$$\begin{aligned}
&(\|x_e^{k+1}\|_{Q_1}^2 + \|z_e^{k+1}\|_{Q_2}^2 + \alpha\|z_e^{k+1}\|^2 + \|y_e^{k+1}\|_C^2) \\
&\quad \leq \|x_e^k\|_{Q_1}^2 + \|z_e^k\|_{Q_2}^2 + \alpha\|z_e^k\|^2 + \|y_e^k\|_C^2.
\end{aligned}$$

Since the matrices Q_1 and C are positive definite, the sequence (x^k, z^k, y^k) is then bounded. By summing (4.10) from 0 to ∞, we can also derive

$$\sum_{k=0}^{\infty}(\|x^{k+1} - x^k\|_{Q_1}^2 + \|z^{k+1} - z^k\|_{Q_2}^2) < \infty, \qquad \sum_{k=0}^{\infty}\|Bx_e^{k+1} - z_e^k\|^2 < \infty$$

and

$$\sum_{k=0}^{\infty}\langle t_e^{k+1}, x_e^{k+1}\rangle < \infty, \qquad \sum_{k=0}^{\infty}\langle s_e^{k+1}, t_e^{k+1}\rangle < \infty.$$

Thus the equalities

$$\lim_k \|Bx_e^{k+1} - z_e^{k+1}\| = \lim_k \|Bx^{k+1} - z^{k+1}\| = 0$$

and

$$\lim_k H(x^k) = H(\overline{x}); \qquad \lim_k J_1(z^k) = J_1(\overline{z})$$

hold. By the boundness of the sequence, there exists convergent subsequence of (x^k, z^k, y^k), and each cluster point $(x^\infty, z^\infty, y^\infty)$ satisfies

$$\begin{cases} t^\infty + B^T y^\infty = 0, \\ s^\infty - y^\infty = 0, \\ Bx^\infty - z^\infty = 0. \end{cases} \tag{4.11}$$

Thus $(x^\infty, z^\infty, y^\infty)$ is a saddle point of $L_1(x, z;\ y)$ and (x^∞, z^∞) is a solution of (2.6). □

Remark 4.3 For algorithm (A_0), if the functional $J_0(x) + \frac{\alpha}{2}\|Ax - b\|^2$ is strictly convex with respect to x, the positive definitive condition for the matrix Q_0 can be relaxed to semi-positive. Also, if the functional $H(x) + \frac{\alpha}{2}\|Bx - z^k\|^2$ is strictly convex with respect to x for algorithm (A_1), then the condition for Q_1 can be relaxed to positive semi-definite. The proofs follows the ones of Theorems 4.1 and 4.2.

5 Applications

In this section, we consider the applications of the algorithms (A_0) and (A_1) in signal and image processing. We will present new algorithms and also draw the connection to some existing methods. Throughout, we choose

$$C = \frac{1}{\gamma} I,$$

where $\gamma > 0$ for both algorithms and $Q_2 = 0$ for the algorithm (A_1) since Q_2 can be positive semi-definite.

5.1 ℓ^1 Basis Pursuit

The basis pursuit problem [19] solves the constrained minimization problem

$$\text{(Basis Pursuit)} \quad \min_{x \in \mathbb{R}^N} \|x\|_1 \quad \text{s.t. } Ax = b. \tag{5.1}$$

Since A is often under-determined, the linear equation $Ax = b$ has more than one solution. This problem arises in many applications of compressive sensing (CS), for which some of the original work was done in [13] by Candes et al. and in [23] by Donoho. The fundamental principle of CS is that if a signal is sparse in a chosen basis, the signal can be recovered through convex optimization with very few measurements. There are extensive references in this domain. See [50] and references therein.

To be consistent with the parameters in [50] and for linearized Bregman in [42], we choose $J(x)=\mu\|x\|_1$ with a positive parameter $\mu>0$ and set $\alpha=1$. The generic algorithm (A_0) applied to this problem is written as

$$\begin{cases} x^{k+1}=\arg\min_x(\mu\|x\|_1-\langle y^k, Ax\rangle+\frac{1}{2}\|Ax-b\|^2+\frac{1}{2}\|x-x^k\|_{Q_0}), \\ y^{k+1}=y^k+\gamma(b-Ax^{k+1}). \end{cases} \tag{5.2}$$

- If $Q_0=0$, and $\gamma=1$, then

$$\begin{cases} x^{k+1}=\arg\min_x(\mu\|x\|_1+\frac{1}{2}\|Ax-(y^k+b))\|^2), \\ y^{k+1}=y^k+(b-Ax^{k+1}). \end{cases} \tag{5.3}$$

With a change of variable, we obtain the equivalent Bregman iteration for ℓ^1 basis pursuit used in [50]. Note that even though Q_0 is not positive definite in this case, and $J(u)$ is not strictly convex, the convergence of the above algorithm is still established by the equivalence to method of multipliers. In general, solving the first subproblem exactly requires infinite inner iterations. In particular, in [50], the first subproblem is solved by the Fixed Point Continuation (FPC) method proposed by Hale, Yin and Zhang [34]. The idea is based on forward-backward operator splitting (1.8) and a dynamic μ. Note that the first subproblem can be also solved by other methods, such as the iterative shrinkage by Daubechies, De Frise and De Mol in [22, 28], ℓ^1_ls [36], a greedy algorithm [38], FISTA [3] and NESTA [4].

- Another popular ℓ^1 basis pursuit algorithm called linearized Bregman starts with $x^0=0$, $y^0=b$ and updates x^{k+1} with

$$\begin{cases} x^{k+1}=\min_x\mu\|x\|_1+\frac{1}{2\delta}\|x-A^\top y^{k+1}\|^2, \\ y^{k+1}=y^k+\delta(b-Ax^{k+1}). \end{cases} \tag{5.4}$$

This algorithm can be derived by applying the classical Uzawa method (2.4) to the approximated problem

$$x=\arg\min\left(\mu\|x\|_1+\frac{1}{2\delta}\|x\|^2\right) \quad \text{s.t. } Ax=b.$$

Since the functional is strictly convex, the convergence can be also guaranteed. To make it equivalent to the original problem, we need to choose a sufficiently large μ. The equivalence of solutions of this problem and the original basis pursuit problem is demonstrated in [49]. On the other hand, a large μ leads to a slow convergence. An acceleration technique called "kicking" is used in [42].

- Based on a similar idea as linearized Bregman, we want to be able to explicitly solve for the updates in every iteration. By applying the algorithm A_0 (3.2) with $Q_0=\frac{1}{\delta}-A^\top A$, we obtain the proposed inexact Uzawa algorithm (which is equivalent to BOS in [51]): $x^0=0$, $y^0=0$,

$$\begin{cases} x^{k+1}=\arg\min_u(\mu\|x\|_1+\frac{1}{2\delta}\|x-((1-\delta A^\top A)x^k+\delta A^\top(y^k+b))\|^2), \\ y^{k+1}=y^k+\gamma(b-Ax^{k+1}). \end{cases} \tag{5.5}$$

Algorithm 1 Inexact Uzawa Method for ℓ^1 basis pursuit

```
Require: Initialization: $x^0 = 0$, $y^0 = b$.
  while $\|b - Ax^k\| \geq btol$ and $k < K$ do
    $x^{k+1} = \text{shrinkage}(x^k + \delta A^\top(y^k - Ax^k), \mu\delta)$;
    $y^{k+1} = y^k + \gamma(b - Ax^{k+1})$;
  end while
```

This is equivalent to : $x^0 = 0$, $y^0 = b$:

$$\begin{cases} z^{k+1} = x^k - \delta A^\top(Ax^k - y^k)), \\ x^{k+1} = \text{Shrinkage}(z^{k+1}, \mu\delta), \\ y^{k+1} = y^k + \gamma(b - Ax^{k+1}). \end{cases} \tag{5.6}$$

This algorithm can be also interpreted as PCPM (2.16), and each step can be computed explicitly. Note that this formulation can be viewed as a one-step inner iteration of PFBS with outer Bregman iterations. A similar idea is also applied in [5] for parallel compressive sensing, where a continuation technique is used instead of Bregman iterations.

Finally, the proposed inexact Uzawa numerical scheme (5.5) is presented in Algorithm 1.

In the following, we demonstrate the numerical performance of the proposed algorithm. We will also compare it with linearized Bregman and PFBS + Bregman with additional inner iterations. We generate sparse signals $\overline{x}$ of length n with different settings. More precisely, we fix the length of the measurements b to be $m = n/2$, and the number of non-zeros of the signals equal to ρm where $\rho = 0.02$ and $\rho = 0.2$. The stopping tolerance is $btol = \frac{\|Ax^k - b\|}{\|b\|} \leq 10^{-5}$ and the maximum number of iterations is 1000. Two kinds of standard compressive sensing measurements matrices are used: orthogonalized Gaussian matrices and partial discrete cosine transform (DCT). For the random Gaussian matrices, the elements are generated from i.i.d normal distributions $N(0, 1)$ and the rows of A were orthogonalized. For the DCT measurements, we use the fast DCT transform without explicitly storing the measurement matrix.

According to Theorem 4.1: the parameter δ must be chosen so that the matrix $Q_0 = \frac{1}{\delta} - A^\top A$ is positive definite, that is $0 < \delta < \frac{1}{\|A^\top A\|}$. In the case of compressive sensing, we can choose $0 < \delta < 1$. The parameter γ is a step-size for the gradient ascent step for the dual variable. The convergence of the algorithm (A_0) is guaranteed when $0 < \gamma < 2$ $(\alpha = 1)$, while in linearized Bregman and PFBS + Bregman iterations this parameter γ is fixed as 1. Numerically, we observe that a larger γ generally yields a faster convergence while the residual converges to zero in an oscillating fashion. In general $\gamma = 1$ gives a good convergence rate. Thus, we fix $\gamma = 1$ and $\delta = 0.99$ for the inexact Uzawa method. We use the relative error (RelErr): $\frac{\|x^k - \overline{x}\|}{\|\overline{x}\|}$ to measure reconstruction quality.

In Fig. 1, we show the residual and the corresponding RelErr for the sequence generated by the inexact Uzawa method. Theoretically, the residual $\|Ax^k - b\| \to 0$, but the residual is not monotonically decreasing. This is due to the fact that we are not minimizing the quadratic term $\|Ax - b\|^2$ in one iteration. To avoid this and keep a relatively faster convergence, we only record a subsequence of x^k which has decreasing residual, see Fig. 1. Note that this treatment will not change or accelerate the algorithm, and by the convergence result (Theorem 4.1) this subsequence will converge to an optimal solution. In the following, we use the subsequence for the inexact Uzawa method.

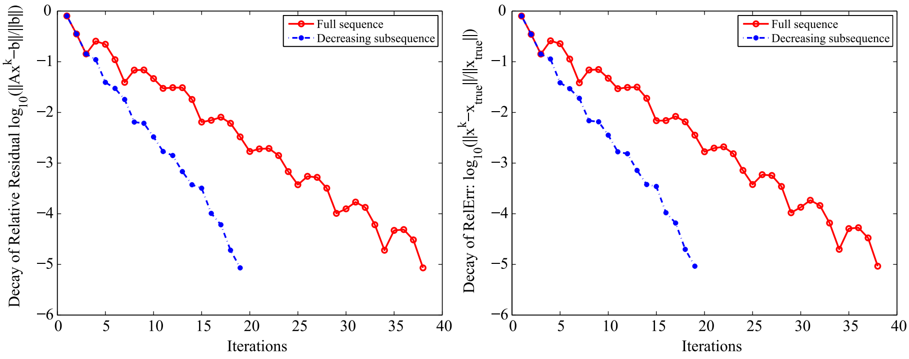


Fig. 1 Residual decay (*left*) and RelErr decay (*right*) of full sequence and residual-decreasing subsequence of inexact Uzawa method. DCT measurements for $n = 1024$, $m = 512$, $\rho = 0.02$, $\gamma = 1$, $\delta = 0.99$, $\mu = 1$

Now we compare our algorithm to linearized Bregman and PFBS + Bregman. Note that we do not compare to other more complicated implicit algorithms for solving the first subproblem, such as FPC [34] and methods based on Nestorov's approach such as Fista [3], Nesta [4]. The efficiency of these algorithms depends on more sophisticated parameter selections and stopping criterions. Among the three algorithms we are comparing, FFBS + Bregman has one more parameter for the number of inner iterations. In our experiments, we fix this number as $nInner = 10$. For both linearized Bregman and PFBS + Bregman, we choose $\delta = 1.99$ since we can choose $0 < \delta < \frac{2}{\|A^\top A\|}$ and they achieve the best convergence rate when δ is close to the upper bound. The most important parameter is μ. The total number of iterations and running time of these three algorithms closely depend on μ. As we mentioned previously, a large μ is preferred for linearized Bregman, so as not to get a wrong solution. For the reason of comparison, we adopt the method of linearized Bregman with kicking proposed in [42]. The idea is to jump unchanged steps due to a big shrinkage in order to accelerate the convergence.

In Table 1, we show the behaviors of the three algorithms with different choices of parameter μ under two experimental setting: sparsity $\rho = 0.02$ and $\rho = 0.2$. For each setting, the first row denotes the linearized Bregman result, the second row is for the inexact Uzawa method and the last row is for PFBS + Bregman. We can see from this table that when μ ($\mu = 0.1$) is small, linearized Bregman can fail ($RelErr = 0.25$) even when the residual decreases to 0. This could be severe for the less sparse case ($\rho = 0.2$, $RelErr = 0.61$). In contrast, for PFBS + Bregman, a smaller μ leads to faster convergence and higher accuracy rate. When μ is large ($\mu = 10$) and $\rho = 0.2$, the convergence is very slow since a lot of sub-steps are wasted. Among these three algorithms, the inexact Uzawa method is relatively more robust to μ compared to linearized Bregman and PFBS + Bregman. In the following, we fix $\mu = 0.1$ for the inexact Uzawa method and PFBS + Bregman, $\mu = 1$ for $\rho = 0.02$, and $\mu = 10$ for $\rho = 0.2$ respectively for linearized Bregman with kicking. The comparisons of these three methods for different sizes of problems and different measurement matrices are presented in Tables 2 and 3. We can observe that with the best μ for each method, linearized Bregman converges faster than the inexact Uzawa method for the case of very sparse signals, while the inexact Uzawa method is faster when signals are denser. The PFBS + Bregman method is in general slower than the other two methods, but it can achieve a slightly lower Relerr rate for most of cases.

Table 1 Experimental results using 20 DCT runs for $n = 1024$, $m = n/2$, $k = \rho m$, $\rho = 0.02$ and $\rho = 0.2$. For each experiment, the first row is the result with linearized Bregman with kicking and the second row is the inexact Uzawa method, and the last row is PFBS with Bregman iterations (*nInner* = 10)

	$\rho = 0.02$			$\rho = 0.2$		
	RelRes	RelErr	Time	RelRes	RelErr	Time
$\mu = 0.1$	9.9e−6	0.25	0.88	1e−5	0.61	1.1
	7.4e−6	7.8e−6	0.043	7.2e−6	1.1e−5	0.094
	2.1e−6	2.1e−6	0.016	7.2e−6	1.1e−5	0.31
$\mu = 1$	4.6e−6	4.9e−6	0.018	1e−5	0.082	2.4
	6.7e−6	7.1e−6	0.044	8.4e−6	1.2e−5	0.32
	2.6e−6	2.7e−6	0.13	2.7e−6	3.2e−5	1.4
$\mu = 10$	4.8e−6	5.1e−6	0.037	7.4e−6	1.1e−5	0.27
	8e−6	8.5e−6	0.2	1.1e−5	1.5e−5	2.7
	2.2e−6	2.3e−6	1.2	1.5e−3	1.7e−3	3.4

Table 2 Experimental results using 20 DCT runs for $n, m, k = \rho m$. For each experiment, the first row is the result with linearized Bregman with kicking ($\mu = 1$ for $\rho = 0.02$ and $\mu = 10$ for $\rho = 0.2$), the second row is the inexact Uzawa algorithm proposed ($\mu = 0.1$ for both cases) and the last row is PFBS with Bregman iterations ($\mu = 0.1$, *nInner* = 10 for both cases)

n	m	ρ	RelErr			Time		
			mean	std	max	mean	std	max
1024	512	0.02	4.9e−6	2.4e−6	9.1e−6	0.014	0.007	0.031
			7.8e−6	2.8e−6	1.1e−5	0.042	0.01	0.062
			4.4e−6	2.7e−6	9.8e−6	0.029	0.014	0.047
4096	2048	0.02	4.1e−6	2.3e−6	1.1e−5	0.094	0.033	0.16
			9.2e−6	1.2e−6	1e−5	0.17	0.082	0.51
			5.1e−6	2.6e−6	9.7e−6	0.33	0.66	3.1
32,768	16, 384	0.02	4.8e−6	2.8e−6	9.5e−6	1.9	0.21	2.2
			8.6e−6	2.7e−6	1.1e−5	1.8	1.3	6.6
			4.9e−6	2.8e−6	1e−5	12	21	97
1024	512	0.2	1.1e−5	2.2e−6	1.5e−5	0.28	0.051	0.41
			1.1e−5	2.7e−6	1.6e−5	0.1	0.068	0.37
			9.9e−6	2.1e−6	1.3e−5	0.27	0.46	2.1
4096	2048	0.2	9.8e−6	2.3e−6	1.5e−5	2.1	0.38	3.1
			1.1e−5	3e−6	1.5e−5	0.52	0.38	1.9
			8.9e−6	2.2e−6	1.3e−5	2.1	2.3	10
32,768	16, 384	0.2	9.8e−6	2.2e−6	1.2e−5	59	5.6	69
			9.2e−6	2.2e−6	1.4e−5	6.8	2.2	12
			9.2e−6	2.2e−6	1.1e−5	30	9.8	46

Table 3 Experiment results using 20 Gaussian runs for different $n, m, k = \rho m$. For each experiment setting, the first row is with linearized Bregman with kicking ($\mu = 1$ for $\rho = 0.02$ and $\mu = 10$ for $\rho = 0.2$), the second row is the inexact Uzawa algorithm proposed ($\mu = 0.1$ for both cases) and the last row is PFBS + Bregman iterations ($\mu = 0.1$, *nInner* = 10 for both cases)

n	m	ρ	RelErr			Time		
			mean	std	max	mean	std	max
1024	512	0.02	3.6e−6	2e−6	7.6e−6	0.12	0.038	0.2
			7.6e−6	2.6e−6	1.1e−5	0.22	0.045	0.31
			3.8e−6	2e−6	7.8e−6	0.1	0.037	0.19
4096	2048	0.02	5e−6	2.9e−6	1.1e−5	1.6	0.37	2.5
			7.9e−6	1.7e−6	1e−5	2.8	0.96	6.8
			4.9e−6	2.2e−6	8.6e−6	6.1	14	65
1024	512	0.2	1.2e−5	2.5e−6	1.9e−5	1.3	0.31	2.3
			1.2e−5	2.4e−6	1.5e−5	0.47	0.3	1.4
			9.7e−6	2.8e−6	1.5e−5	1.5	3.1	12
4096	2048	0.2	1e−5	2.2e−6	1.4e−5	34	6.1	48
			1.2e−5	2.5e−6	1.6e−5	8.3	6.6	31
			9.2e−6	2.2e−6	1.4e−5	33	38	160

For the noisy case, the optimization model is different. We can also apply the inexact Uzawa method for the related primal-dual formulation. This example is discussed in a continued work [27].

5.2 TV-L^2 Regularization

Now we demonstrate an application of problem (2.6) with separable structure. We consider TV-L^2 minimization problems in an unconstrained form,

$$\min_u \left(\mu \mathrm{TV}(x) + \frac{1}{2}\|Ax - b\|^2\right) \tag{5.7}$$

where $\mathrm{TV}(x) = \sum_{i,j} \|(\nabla x)_{i,j}\|_1$, and $(\nabla x)_{i,j} \in \mathbb{R}^2$ denotes a discrete gradient of x at (i, j). As in [51], the related constrained formulation can be formulated as

$$\min_u \mathrm{TV}(x) \quad \text{s.t. } Ax = b \tag{5.8}$$

and it can be solved either by Algorithm (A_0) or (A_1). Similar to the ℓ^1 basis pursuit problem, the algorithm (A_0) applied to (5.8) is BOS as presented in [51], with the ℓ^1 minimization step replaced by a ROF [45] denoising step. Some efficient algorithms, such as graph cuts [21] or the dual projection method of Chambolle [15] can be applied.

Here, we are more interested in solving the unconstrained problem (5.7) by the algorithm (3.3). Following the split Bregman method proposed in[32], we first reformulate the problem (5.7) as

$$\min_{x,z}\left(\mu\|z\|_1 + \frac{1}{2}\|Ax - b\|^2\right) \quad \text{s.t. } \nabla x = z$$

The Algorithm (A_1) applied to the unconstrained problem (5.7) is written as $x^0=0$, $z^0=0$, $y^0=0$,

$$\begin{cases} x^{k+1}=\arg\min_x(\frac{1}{2}\|Ax-b\|^2+\langle y^k,\nabla x\rangle+\frac{\alpha}{2}\|\nabla x-z^k\|^2+\frac{1}{2}\|x-x^k\|^2_{Q_1}), \\ z^{k+1}=\arg\min_z(\mu\|z\|_1-\langle y^k,z\rangle+\frac{\alpha}{2}\|\nabla x^{k+1}-z\|^2+\frac{1}{2}\|z-z^k\|^2_{Q_2}), \\ Cy^{k+1}=Cy^k+(\nabla x^{k+1}-z^{k+1}). \end{cases} \tag{5.9}$$

Since we choose $Q_2=0$ and $C=\frac{1}{\gamma}$ for simplicity, we obtain

$$\begin{cases} x^{k+1}=(A^\top A+Q_1-\alpha\Delta)^{-1}(A^\top b+\mathrm{div}(y^k-\alpha z^k)+Q_1x^k), \\ z^{k+1}=\mathrm{shrinkage}(\nabla x^{k+1}+\frac{1}{\alpha}y^k,\frac{\mu}{\alpha}), \\ y^{k+1}=y^k+\gamma(\nabla x^{k+1}-z^{k+1}) \end{cases} \tag{5.10}$$

where Δ denotes discrete Laplace operator and div denotes the divergence operator, the adjoint of the gradient operator. The variants of algorithms then depend on how we choose the matrix Q_1.

- Let $Q_1=\frac{1}{\delta}-A^\top A+\alpha\Delta$, then x^{k+1} is updated as

$$x^{k+1}=x^k-\delta A^\top(Ax^k-b)+\delta\mathrm{div}(\alpha\nabla x^k-\alpha z^k+y^k). \tag{5.11}$$

  In this case, the step corresponds to a one-step gradient descent and it is very efficient since it doesn't involve any operator inverting. We notice that this is an *explicit* algorithm. Note that δ must be chosen such that Q_1 is positive definite, therefore $0<\delta<\frac{1}{\|A^\top A-\alpha\Delta\|}$.
- If A is not diagonalizable, we can choose Q_1 as $Q_1=\frac{1}{\delta}-A^\top A$. Then x^{k+1} is solved by

$$x^{k+1}=\left(\frac{1}{\delta}-\alpha\Delta\right)^{-1}\left(\left(\frac{1}{\delta}-A^\top A\right)x^k+A^\top b+\mathrm{div}(y^k-\alpha z^k)\right). \tag{5.12}$$

  We refer to this algorithm as a *semi-implicit* variant. According to Theorem 4.2, the condition for δ is $0<\delta<\frac{1}{\|A^\top A\|}$.
- If A is diagonalizable in Fourier basis or DCT depending on the boundary conditions, then $A^\top A-\alpha\Delta$ is diagonalizable. We can choose $Q_1=\frac{1}{\delta}$. Then x^{k+1} is solved by

$$x^{k+1}=\left(A^\top A-\alpha\Delta+\frac{1}{\delta}\right)^{-1}\left(\frac{1}{\delta}x^k+A^\top b+\mathrm{div}(y^k-\alpha z^k)\right). \tag{5.13}$$

  This is exactly the alternating split Bregman method proposed by Goldstein and Osher in [32]. The condition for δ is $0<\delta<\infty$. We refer to this as an *implicit* algorithm.

Finally, note that the three algorithms can be written in one unified scheme as Algorithm 2.

In the following, we provide an example of TV-L^2 for Computerized Tomography (CT). In a simplified parallel tomographic problem, an observed body slice is modeled as a two-dimensional function, and projections modeled by line integrals represents the total attenuation of a beam of x-rays when it traverses the object. The acquired projections are parameterized by the angle of each beam and the signed perpendicular distance from each line to the origin. The projections can be represented by the Radon transform, and the tomographic reconstruction problem is then to estimate the function from a finite number of measured line integrals, see [2] for more background. The standard reconstruction algorithm in clinical applications is Filtered Back Projection (FBP). In the presence of noise, this problem

Algorithm 2 Generalized Split Bregman Method for $TV - L^2$ regularization

```
Require: Input parameters $\mu, \delta, \gamma, \alpha, btol, K$.
  Initialization: $x^0 = 0, y^0 = b, k = 0, r_0 = b - Ax^0$.
  while $\|r_0\| \leq btol$ and $k < K$ do
    if mode is ‘explicit’ then
      Update $x^{k+1}$ by (5.11)
    else if mode is ‘semi-implicit’ then
      Update $x^{k+1}$ by (5.12)
    else
      Update $x^{k+1}$ by (5.13){mode is ‘explicit’}
    end if
    $z^{k+1} = \text{shrink}(\nabla x^{k+1} + \frac{1}{\alpha} y^k, \frac{\mu}{\alpha})$.
    $y^{k+1} = y^k + \gamma(\nabla x^{k+1} - z^{k+1})$
  end while
```

becomes difficult since the inverse of the Radon transform is unbounded. Therefore, we use total variation as a regularizer. The model we consider takes the form (5.7) with A being the Radon transform matrix. We simulate a 128 × 128 Shepp-Logan phantom image and its 50 uniformly oriented projections with Gaussian noise. For this case, it is easier to compute the adjoint instead of inverse of the Radon transform, thus we only consider the semi-implicit and explicit modes. The results are shown in Fig. 2. Both methods are very efficient and obtain similar results. Figure 3 shows the evolution of the energy for the two algorithms. As we expect, the semi-implicit mode converges in fewer iterations than the explicit method, but it takes more time for each iteration. They take roughly the same amount of time to achieve similar results.

6 Extensions

6.1 Matrix Completion

A related problem to compressive sensing is matrix completion. The matrix completion problem consists of recovering a low rank matrix X from its incomplete known entries. However, a rank minimization problem is in general NP hard. In [12], Candès and Recht showed that some low rank matrices can be recovered exactly from an incomplete sampling by solving a convex optimization problem:

$$\min_X \|X\|_* \quad \text{s.t. } P_\Omega(X) = b \tag{6.1}$$

where $\|X\|_*$ is the nuclear norm of X defined as the sum of singular values of X, Ω is the index set of known entries, P_Ω is the projection operator, and b is the vector of known entries.

If we denote $\langle X, Y\rangle = \text{trace}(X^*Y)$ and $\|X\|_F^2 = \langle X, X\rangle$, a key property presented in [8] is based on the proximity operator of the nuclear norm:

$$\arg\min_X \left(\tau\|X\|_* + \frac{1}{2}\|X - Y\|^2\right) = \text{Shrinkage}(Y, \tau) \tag{6.2}$$

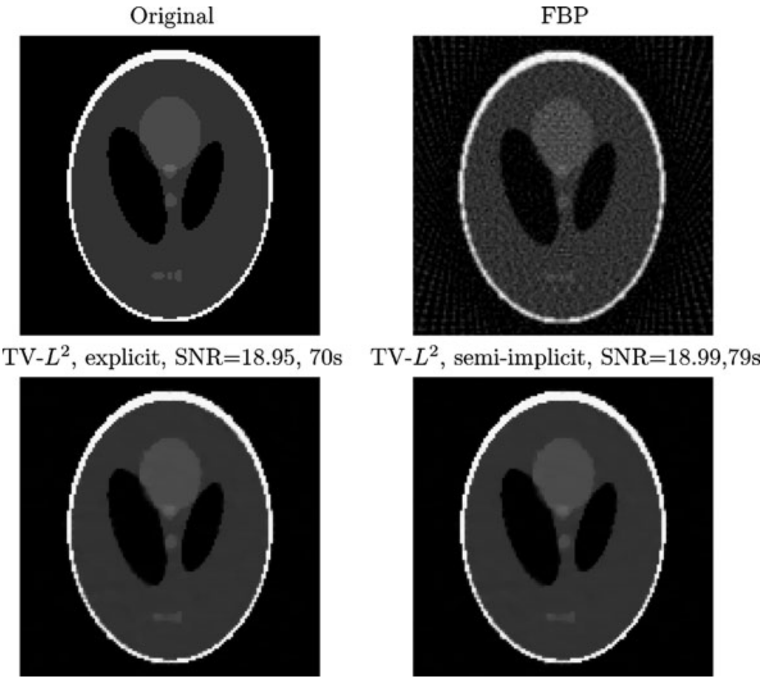
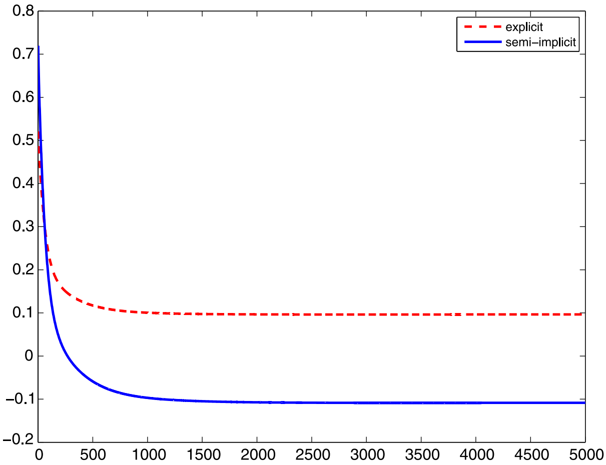

Original
FBP
TV-L^2, explicit, SNR=18.95, 70s
TV-L^2, semi-implicit, SNR=18.99,79s

Fig. 2 A tomographic reconstruction example for a 12 × 128 image, with 50 projections. The initial guess for both explicit and semi-implicit methods is the FBP image

Fig. 3 $\log_{100}$(energy) *vs* iterations of explicit and semi-implicit methods for the above example



where Shrinkage(Y, τ) is soft-thresholding with parameter τ of the singular values of the matrix Y. More precisely, consider the singular value decomposition of a matrix $Y \in \mathbb{R}^{n_1 \times n_2}$ with rank r: $Y = U\Sigma V^*$, $\Sigma = \text{diag}(\{\sigma_i\}_{i=1}^r)$ where U, V are respectively $n_1 \times r$ and $n_2 \times r$ matrices with orthonormal columns, and σ_i are positive singular values. The analogue of soft-shrinkage of a signal (1.9), soft-shrinkage of matrix Y is defined as

$$\text{Shrinkage}(Y, \tau) := U\text{diag}(\{\max(\sigma_i - \tau, 0)\})V^*. \tag{6.3}$$

Now we can use this proximal point minimization of the nuclear norm to solve the matrix completion problem. We first write the primal-dual formulation of (6.1) as

$$L(X,p)=\mu\|X\|_*+\frac{1}{2}\|P_\Omega(X)-b\|^2+\langle P_\Omega(X)-b,p\rangle. \tag{6.4}$$

Similar to the basis pursuit problem, a Singular Value Thresholding (SVT) algorithm, based on linearized Bregman, is proposed in [8]. Fixing $\tau>0$ and a positive number δ (or a sequence $\{\delta_k\}_{k\geq 0}$), the algorithm is as follows

$$\begin{cases} X^{k+1}=\mathrm{Shrinkage}(P_\Omega^\top Y^k,\tau),\\ Y^{k+1}=Y^k+\delta(b-P_\Omega X^{k+1}). \end{cases} \tag{6.5}$$

In [8], the authors have proved that the algorithm (6.5) is an application of the classic Uzawa method to the convex optimization problem:

$$\min \tau\|X\|_*+\frac{1}{2}\|X\|_F^2,\quad \text{s.t. } P_\Omega(X)=b.$$

Due to the strict convexity of the approximated functional, the algorithm converges to a unique solution.

Our analogous algorithm for the matrix completion problem is derived in the same way as ℓ^1 basis pursuit. The idea is to solve the original problem (6.1) by using the inexact Uzawa method (3.2)

$$\begin{cases} X^{k+1}=\mathrm{Shrinkage}(X^k-\delta P_\Omega^T(P_\Omega X^k-Y^k),\delta\mu),\\ Y^{k+1}=Y^k+\gamma(b-P_\Omega(X^{k+1})). \end{cases} \tag{6.6}$$

The convergence of this algorithm follows from the proof of Theorem 4.1 and Theorem 4.2 in [8]. Another algorithm we consider is the Bregman iterations method with possibly more than one iteration of PFBS solving the subproblem, that is

$$\begin{cases} \begin{cases} Z^{k+1}=X^k-\delta P_\Omega^T(P_\Omega X^k-Y^k),\\ X^{k+1}=\mathrm{Shrinkage}(Z^{k+1},\delta\mu), \end{cases}\\ Y^{k+1}=Y^k+\gamma(b-P_\Omega(X^{k+1})). \end{cases} \tag{6.7}$$

Note that this is a simplified version of the fixed point continuation (FPC) method with Bregman iterations in [39], where a dynamic parameter μ is used in addition to the Bregman iterations.

In the following, we compare the performance of the three methods: SVT (linearized Bregman) (6.5), the inexact Uzawa method (6.6) and Bregman iterations with PFBS (6.7). As in [8], we simulate $n\times n$ matrices of rank r by multiplying two random matrices of size $n\times r$ and then the given entries support Ω of cardinality m are uniformly randomly sampled. We use the same stopping criterion as in [8], that is $\frac{\|P_\Omega(X^k)-b\|}{\|b\|}\leq 10^{-4}$. We first look at the behavior of the inexact Uzawa method (6.6) with different μ. Figure 4 shows the evolution of the residual $\|P_\Omega(X^k)-b\|$ and the rank of X^k. We can see that the residual decreases faster with a smaller μ. The behavior of the rank is different. With a smaller μ, the rank of X^k is big at the beginning and it starts to decrease afterward. With a larger μ the rank increases and stabilizes. Thus a smaller μ will result in a large rank matrix at the beginning. It requires more singular value decomposition computation, even though it converges in

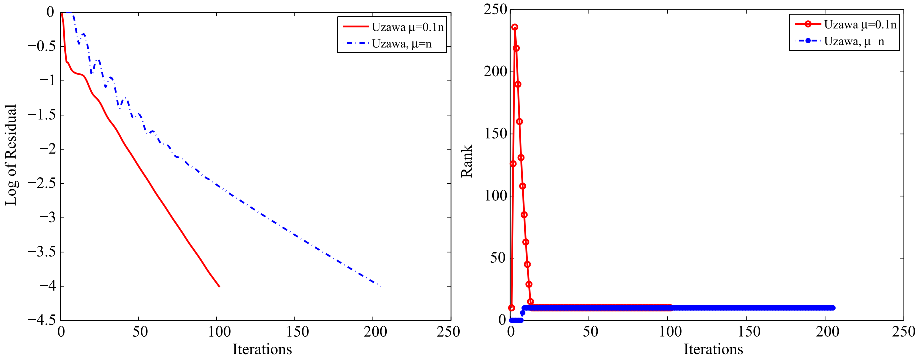


Fig. 4 Matrix Completion by inexact Uzawa method with different $\mu = n$ and $\mu = \frac{n}{10}$ for $n = 1000$, $r = 10$, $m = 0.12 * n^2$, $\delta = 1$

Table 4 Experiment results using 5 runs for different matrices of size $n \times n$, rank r, and sampling rate $p = m/n^2$. The 1st row is linearized Bregman with $\delta = 1.2/p$, $\mu = 5n/\delta$. The 2nd row is the inexact Uzawa algorithm with $\delta = 1$, $\mu = n$. The last row is PFBS(10 inner steps) + Bregman with $\delta = 1$, $\mu = n/5$

n	r	p	RelErr	Time	Steps
1000	10	0.12	1.637e−4	154.3	119
			1.666e−4	224.7	202.4
			1.438e−4	202.4	240.6
1000	50	0.39	1.628e−4	344.4	116
			1.55e−4	109.5	61.6
			1.147e−4	320.4	105.4
1000	100	0.57	1.703e−4	631.5	130
			1.53e−4	197.1	48
			1.349e−4	1144	85

fewer iterations. In fact, for the example in Fig. 4, the computation times are 407 s and 101 s for $\mu = 0.1n$ and $\mu = n$ respectively. We also want to compare the performance of the three methods mentioned above. Table 4 shows the results. As in the ℓ^1 compressive sensing case, the parameter μ needs to be big for the SVT method. We use the proposed $\mu\delta = 5n$ in [8] for SVT, $\delta = 1$ and $\mu = n$ for the Uzawa method, and $\delta = 1$, $\mu = 0.2n$ for the PFBS + Bregman with $n = 10$ inner steps. We can see that the inexact Uzawa method is slower than linearized Bregman and PFBS + Bregman when the rank is low ($n = 1000$, $r = 10$), but it is faster than the other two methods when the rank is higher ($n = 1000$, $r = 50$, $r = 100$).

7 Conclusions

We have proposed two general primal-dual methods for solving linearly constrained problems arising from signal and image processing applications. The algorithms generalize Bregman operator splitting (BOS) proposed in [51] and are connected to other popular methods, such as linearized Bregman, split Bregman methods and PCPM. The numerical examples show the proposed algorithms are efficient, robust to the parameter selection and flexible enough to be widely applicable. In theory, our algorithms can be extended to minimize other

TV or ℓ^1 based functionals, such as TV$-L^1$ as well as more general primal dual formulations of convex programming involving more complicated constraint sets. The code for the applications presented in this paper will be available online.

Acknowledgements X. Zhang would like to thank Ernie Esser for fruitful discussions and valuable suggestions to improve the quality of the paper. M. Burger and S. Osher thank Fondazione CIME for a summer school in stimulating atmosphere, initiating a part of this project. The work is supported by NSF CCF-0528583, NSF DMS-0610079 and NSF DMS-0312222, ONR-N00014-08-1-119, ONR-N00014-07-1-0810, NSF DMS-0914561, NSF-DMS-0835863.

References

1. Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in Linear and Non-Linear Programming. Stanford University Press, Stanford (1958)
2. Avinash, C., Malcolm, S.: Principles of Computerized Tomographic Imaging. Society for Industrial and Applied Mathematics, Philadelphia (2001)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 693–696 (2009)
4. Becker, S., Bobin, J., Candès, E.: Nesta: a fast and accurate first-order method for sparse recovery (2009)
5. Borghi, A., Darbon, J., Peyronnet, S., Chan, T.F., Osher, S.: A simple compressive sensing algorithm for parallel many-core architectures. CAM Report 08-64, UCLA (2008)
6. Bramble, J.H., Pasciak, J.E., Vassilev, A.T.: Analysis of the inexact Uzawa algorithm for saddle point problems. SIAM J. Numer. Anal. **34**(3), 1072–1092 (1997)
7. Bregman, L.: The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. USSR Comput. Math. Math Phys. **7**, 200–217 (1967)
8. Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
9. Cai, J.-F., Osher, S., Shen, Z.: Convergence of the linearized Bregman iteration for ℓ^1-norm minimization. Math. Comput. **78**, 2127–2136 (2009)
10. Cai, J.-F., Osher, S., Shen, Z.: Split Bregman method and frame based image restoration. CAM Report 09-28, UCLA, March 2009
11. Cai, J.-F., Osher, S., Shen, Z.: Linearized Bregman iterations for frame-based image deblurring. SIAM J. Imag. Sci. **2**(1), 226–252 (2009)
12. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. (2008, to appear)
13. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
14. Censor, Y., Lent, A.: An iterative row-action method for interval convex programming. J. Optim. Theory Appl. **34**(3), 321–353 (1981)
15. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imag. Vis. **20**, 89–97 (2004)
16. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. Math. Oper. Res. **219**, 241–252 (1996)
17. Chen, G., Teboulle, M.: Convergence analysis of a proximal like minimization algorithm using Bregman functions. SIAM J. Optim., 3 (1993)
18. Chen, G., Teboulle, M.: A proximal-based decomposition method for convex minimization problems. Math. Program. **64**(1), 81–101 (1994)
19. Chen, S.S., Donoho, David L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998)

20. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Model. Simul. **4**(4) (2005)
21. Darbon, J., Osher, S.: Fast discrete optimization for sparse approximations and deconvolutions. Preprint, UCLA (2007)
22. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. **57**(11), 1413–1457 (2004)
23. Donoho, D.L.: Compressed sensing. IEEE. Trans. Inf. Theory **52**, 1289–1306 (2006)
24. Douglas, J., Rachford, H.H.: On the numerical solutions of heat conduction problems in two and three space variables. Trans. Am. Math. Soc. **1**, 97–116 (1976)
25. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**, 293–318 (1992)
26. Esser, E.: Applications of Lagrangian-based alternating direction methods and connections to split Bregman. CAM Report 09-31, UCLA, April 2009
27. Esser, E., Zhang, X., Chan, T.: A general framework for a class of first order primal-dual algorithms for TV minimization. CAM Report 09-67, UCLA, August 2009
28. Figueiredo, M.A.T., Nowak, R.D., Wright, S.J.: Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE J. Sel. Top. Signal Proces. **1**(4), 586–597 (2007)
29. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems. North-Holland, Amsterdam (1983)
30. Glowinski, R., Le Tallec, P.: Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia (1989)
31. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split Bregman method: segmentation and surface reconstruction. UCLA CAM Report, 09-06, February (2009)
32. Goldstein, T., Osher, S.: The split Bregman method for l^1 regularized problems. SIAM J. Imag. Sci., 2 (2009)
33. Guo, X., Li, F., Michael, K.Ng.: A fast L1-TV algorithm for image restoration. ICM Technical Report 08-13, Hong Kong Baptist University, November 2008
34. Hale, E.T., Yin, W., Zhang, Y.: A fixed-point continuation method for $l1$-regularized minimization with applications to compressed sensing. SIAM J. Optim. **19**, 1107–1130 (2008)
35. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appl. **4**, 303–320 (1969)
36. Kim, S., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale l^1-regularized least squares. IEEE J. Sel. Top. Signal Proces. **1**(4), 606–617 (2007)
37. Lemaréchal, C., Sagastizábal, C.: Practical aspects of the Moreau-Yosida regularization, I: Theoretical properties (1994)
38. Li, Y., Osher, S.: Coordinate descent optimization for ℓ^1 minimization with application to compressed sensing; a greedy algorithm. Inverse Probl. Imaging **3**(3), 487–503 (2009)
39. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. Technical Report, Columbia University (2008)
40. Moreau, J.-J.: Fonctions convexes duales et points proximaux dans un espace hilbertien. C. R. Acad. Sci. Paris Ser. A Math. **255**, 2897–2899 (1962)
41. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation based image restoration. Multiscale Model. Simul. **4**(2), 460–489 (2005)
42. Osher, S., Mao, Y., Dong, B., Yin, W.: Fast linearized Bregman iteration for compressive sensing and sparse denoising. Commun. Math. Sci. **8**(1), 93–111 (2010)
43. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. Optimization (1972)
44. Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res. **82**, 421–439 (1956)
45. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
46. Setzer, S.: Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. In: Morken, K., Lysaker, M., Lie, K.-A., Tai, X.-C. (eds.) Scale Space and Variational Methods in Computer Vision. LNCS, vol. 5567, pp. 464–476. Springer, Berlin (2009)
47. Tai, X.-C., Wu, C.: Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. CAM Report 09-05, UCLA, January 2009
48. Yang, J., Zhang, Y., Yin, W.: An efficient TV-L1 algorithm for deblurring multichannel images corrupted by impulsive noise. SIAM J. Sci. Comput. **31**(4) (2009)
49. Yin, W.: Analysis and generalizations of the linearized Bregman method. CAAM Report, Rice University (2009)

50. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for ℓ^1 minimization with applications to compressed sensing. SIAM J. Imag. Sci. **1**, 143–168 (2008)
51. Zhang, X., Burger, M., Bresson, X., Osher, S.: Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. CAM Report 09-03, UCLA, January 2009
52. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. CAM Report 08-34, UCLA, May 2008