

# LINEARIZED ALTERNATING DIRECTION METHOD FOR CONSTRAINED LINEAR LEAST-SQUARES PROBLEM

RAYMOND H. CHAN\*, MIN TAO†, AND XIAOMING YUAN‡

February 8, 2011

**Abstract.** In this paper, we apply the alternating direction method (ADM) to solve a constrained linear least-squares problem where the objective function is a sum of two least-squares terms and the constraints are box constraints. Using ADM, we decompose the original problem into two easier least-squares subproblems at each iteration. To speed up the inner iteration, we linearize the subproblems whenever their closed-form solutions do not exist. We prove the convergence of the resulting algorithm and apply it to solve some image deblurring problems. We show the efficiency of our algorithm by comparing it with Newton-type methods.

**Key words.** Linear least-squares problems, alternating direction method, linearization, image processing

**AMS subject classifications.** 68U10, 65J22, 65K10, 65T50, 90C25

**1. Introduction.** In this paper, we consider the following constrained linear least-squares problem:

$$\min_{\mathbf{x} \leq \mathbf{u}} \left\{ \frac{1}{2} \|A\mathbf{x} - \mathbf{c}\|^2 + \frac{\lambda^2}{2} \|B\mathbf{x} - \mathbf{d}\|^2 \right\}, \quad (1.1)$$

where  $A, B \in \mathbb{R}^{m \times n}$  with  $m \geq n$ ,  $\mathbf{c}, \mathbf{d} \in \mathbb{R}^m$ ,  $\lambda \in \mathbb{R}$ ,  $\mathbf{l} \in (\mathbb{R} \cup -\infty)^n$  and  $\mathbf{u} \in (\mathbb{R} \cup +\infty)^n$  are given; and  $\|\cdot\|$  denotes the 2-norm. The constraints here are box constraints and are to be interpreted entry-wise, i.e.,  $l_i \leq x_i \leq u_i$  for any  $i \in \{1, 2, \dots, n\}$ .

Clearly (1.1) can be written as  $\min_{\mathbf{x} \leq \mathbf{u}} \frac{1}{2} \|\mathbf{Cx} - \mathbf{e}\|^2$  for some  $C \in \mathbb{R}^{2m \times n}$  and  $\mathbf{e} \in \mathbb{R}^{2m}$ , and there are standard methods such as the Newton-type methods or interior point methods for solving it, see [22, 9, 28]. However, here our emphasis is on applications where  $A$  and  $B$  are two different types of operators and one can design better algorithms to capture their properties. One typical example is in image deblurring, where  $A$  is a blurring operator (integral operator) and  $B$  is a regularization operator (differential operator). When  $\mathbf{d} = \mathbf{0}$ , (1.1) is precisely the Tikhonov regularization with  $\mathbf{c}$  being the observed image,  $\lambda^2$  the regularization parameter, and  $\mathbf{x}$  the image to be restored. The box constraints represent the dynamic range of the image (e.g.,  $l_i = 0$  and  $u_i = 255$  for an 8-bit gray-scale image), see [36]. In the numerical experiments in Section 4, we will see that by imposing the box constraints, one can increase the peak signal-to-noise ratio of the restored images by 0.2 to 2.2 decible. Hence it pays to solve the constrained problem. In addition, (1.1) also serves as one of the two subproblems for the splitting algorithm in [21] which solves the deblurring problem with the total-variational regularization [31]. In that case,  $B = I$  the identity matrix and  $\mathbf{d}$  is an approximation of  $\mathbf{x}$ . We refer to [4, 10] for other applications of (1.1), such as contact problems, control problems, and intensity modulated radiotherapy problems.

In this paper, we develop fast solvers for solving (1.1) that take advantage of the properties of  $A$  and  $B$ . In the literature, algorithms for solving (1.1) are essentially Newton-type methods and are presented

---

\*[rchan@math.cuhk.edu.hk](mailto:rchan@math.cuhk.edu.hk)) Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China. Research is supported in part by HKRGC Grant No. CUHK400510 and CUHK Direct Allocation Grant 2060408.

†[taom@njupt.edu.cn](mailto:taom@njupt.edu.cn)) School of Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. Research is supported by the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (NY210049).

‡[xmyuan@hkbu.edu.hk](mailto:xmyuan@hkbu.edu.hk)) Department of Mathematics, Hong Kong Baptist University, Hong Kong, China. Research is supported by a General Research Fund grant of Hong Kong.

in the context of interior point methods. For example, for solving the nonnegative least-squares problem, some interior point methods were proposed in [9, 28]. In [5, 30], interior point methods have been shown to be particularly tailored for ill-posed problems arising in image reconstruction. In [3], by formulating the Karush-Kuhn-Tucker conditions into a system of nonlinear equations, a Newton-like method was proposed for solving the nonnegative least-squares problem. The authors of [3] applied the precondition technique to deal with the resulting ill-conditioned inner linear system at each Newton iteration when the iterate approaches a solution on the boundary of the feasible set. Their work inspired immediately the reduced Newton method in [23], which solves a subsystem of the inner linear system corresponding only to components of the iterate that are not close to the boundary. Since the resulting subsystem is smaller and less ill-conditioned compared to that in [3], the reduced Newton method outperforms projection type methods and some interior point type Newton methods for image deblurring problems. A most recent approach is the affine scaling method in [7] which solves the Newton steps entry-wise by combining non-monotone line-search strategies and the cyclic version of the classical Barzilai-Borwein stepsize rule [1].

In this paper, we apply the alternating direction method (ADM) proposed originally in [13] to solve the model (1.1). Note that the objective function of (1.1) is a sum of two least-squares terms linked together by a common variable  $\mathbf{x}$ . By introducing an auxiliary variable, we can separate the two least-squares terms and apply ADM directly. ADM decomposes (1.1) into two easier subproblems, each of which is a least-squares problem with only one quadratic term in the objective function. Unless one of the matrices  $A$  and  $B$  is the identity matrix, the resulting ADM subproblems will not have closed-form solutions and will still be difficult to solve. The main contribution of this paper is to apply a linearization technique to solve them. The linearized ADM subproblems are easy to handle as their closed-form solutions can be readily derived. For image deblurring problem, the cost per ADM iteration is about  $O(n \log n)$ , where  $n$  is the number of pixels in the image. We will show that our linearized ADM method converges to the minimizer of (1.1), and we will illustrate the efficiency of the method by testing it on some image deblurring problems and comparing it with existing methods.

The rest of this paper is organized as follows. In Section 2, we present the linearized ADM approach for solving (1.1). In Section 3, we establish the convergence of the derived algorithms. Numerical tests on the new algorithms, including their comparisons with some existing Newton-type methods, are reported in Section 4. Finally, some conclusions are given in Section 5.

**2. The linearized ADM approach.** In this section, we present the linearized ADM approach and derive the algorithms for solving (1.1). We first reformulate (1.1) so that ADM can be applied. Then, we linearize the ADM subproblems so that closed-form solutions can be readily derived.

**2.1. The ADM approach.** Roughly speaking, ADM is an improved variant of the classical augmented Lagrangian method [20, 29] for solving linearly constrained convex programming problems where the objective functions are in a separable form: a sum of individual functions without crossed variables. The efficiency of ADM has been well-documented in the literature, in the particular context of convex programming and variational inequalities, see e.g. [12, 15, 18, 37]. Very recently, ADM has successfully found many applications in the area of image processing [8, 11, 26, 27, 32, 33, 35, 38, 39]. In particular, we refer to [11, 32, 33] for the relationship between ADM and the split Bregman method in [16].

To apply ADM, we first reformulate (1.1) into

$$\min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \Omega} \left\{ \frac{1}{2} \|A\mathbf{x} - \mathbf{c}\|^2 + \frac{\lambda^2}{2} \|B\mathbf{y} - \mathbf{d}\|^2 : \mathbf{x} - \mathbf{y} = 0 \right\}, \quad (2.1)$$

where  $\Omega := \{\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \mid \mathbf{x} \in \mathbb{R}^n\}$  and  $\mathbf{y}$  is an artificial variable. Strictly speaking, (2.1) is a typical linearly-

constrained convex programming problem. Thus, the classical augmented Lagrangian method (ALM) [20, 29] is readily applicable. More specifically, let the augmented Lagrangian function of (2.1) be

$$\mathcal{L}_A(\mathbf{x}, \mathbf{y}, \mathbf{z}, \beta) = \frac{1}{2} \|A\mathbf{x} - \mathbf{c}\|^2 + \frac{\lambda^2}{2} \|B\mathbf{y} - \mathbf{d}\|^2 - \langle \mathbf{z}, \mathbf{x} - \mathbf{y} \rangle + \frac{\beta}{2} \|\mathbf{x} - \mathbf{y}\|^2, \quad (2.2)$$

where  $\mathbf{z} \in \mathbb{R}^n$  is the Lagrange multiplier and  $\beta > 0$  is a penalty parameter for the violation of the linear constraints. Then, with the given  $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{z}^k)$ , the iterative scheme of ALM for (2.1) is

$$\begin{cases} (\mathbf{x}^{k+1}, \mathbf{y}^{k+1}) \leftarrow \arg \min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \Omega} \mathcal{L}_A(\mathbf{x}, \mathbf{y}, \mathbf{z}^k, \beta), \\ \mathbf{z}^{k+1} \leftarrow \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}). \end{cases} \quad (2.3)$$

Note that the direct application of ALM (2.3) treats (2.1) as a generic linearly constrained convex programming problem, without any consideration of its separable structure. Therefore, the variables  $\mathbf{x}^{k+1}$  and  $\mathbf{y}^{k+1}$  are required to be solved simultaneously in (2.3).

The ADM, in contrast, further decomposes the minimization in (2.3) into two subproblems related to  $\mathbf{x}$  and  $\mathbf{y}$ , respectively, and solve them in a consecutive order:

$$\begin{cases} \mathbf{x}^{k+1} \leftarrow \arg \min_{\mathbf{x} \in \mathbb{R}^n} \mathcal{L}_A(\mathbf{x}, \mathbf{y}^k, \mathbf{z}^k), \\ \mathbf{y}^{k+1} \leftarrow \arg \min_{\mathbf{y} \in \Omega} \mathcal{L}_A(\mathbf{x}^{k+1}, \mathbf{y}, \mathbf{z}^k), \\ \mathbf{z}^{k+1} \leftarrow \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}). \end{cases} \quad (2.4)$$

Because of the decomposition, ADM belongs to the class of splitting methods. In fact, the decomposition feature of ADM makes it possible to explore the separable structures of the objective function in (2.1). For example, the  $\mathbf{x}$ -subproblem in (2.4) has a closed-form solution (see (2.5) below).

Now, we delineate the solutions of the two subproblems in (2.4), from which our idea of linearizing these subproblems will become obvious. For the  $\mathbf{x}$ -subproblem in (2.4), it is a least-squares problem without constraints. Hence, it amounts to solving the normal equation:

$$(A^\top A + \beta I)\mathbf{x} = A^\top \mathbf{c} + \mathbf{z}^k + \beta \mathbf{y}^k. \quad (2.5)$$

For image deblurring problems,  $A$  is a spatially-invariant blurring operator. Under the periodic boundary conditions for  $\mathbf{x}$ ,  $A^\top A$  will be a block circulant matrix with circulant blocks, see [6]. Hence it can be diagonalized by the two-dimensional fast Fourier transform (FFT). As a result, (2.5) can be solved exactly using two FFTs (including one inverse FFT), see e.g. [6, 25], and hence the cost is of  $O(n \log n)$ .

On the other hand, it is easy to show that the  $\mathbf{y}$ -subproblem in (2.4) is equivalent to

$$\min_{\mathbf{y} \in \Omega} \left\{ \frac{\lambda^2}{2} \|B\mathbf{y} - \mathbf{d}\|^2 + \frac{\beta}{2} \|\mathbf{y} - \mathbf{x}^{k+1} + \frac{\mathbf{z}^k}{\beta}\|^2 \right\}. \quad (2.6)$$

Its solution is given by the following projection equation:

$$\mathbf{y} = P_\Omega \{ \mathbf{y} - [(\lambda^2 B^\top B + \beta I)\mathbf{y} - (\lambda^2 B^\top \mathbf{d} + \beta \mathbf{x}^{k+1} - \mathbf{z}^k)] \}, \quad (2.7)$$

where  $P_\Omega$  denotes the projection operator onto  $\Omega$  under the Euclidean norm. Obviously, for deblurring problems with the Tikhonov regularization (see e.g. [17]),  $B = I$  in (2.1). Then, the closed-form solution of (2.7) can be easily obtained by:

$$\mathbf{y} = P_\Omega \left[ \frac{1}{\lambda^2 + \beta} (\lambda^2 B^\top \mathbf{d} + \beta \mathbf{x}^{k+1} - \mathbf{z}^k) \right]. \quad (2.8)$$

However, when  $B \neq I$ , the closed-form solution of (2.7) is not available, and we will use linearization to obtain an approximate solution for  $\mathbf{y}$ .

**2.2. The linearized ADM approach for (2.1).** The lack of closed-form solution of (2.6) renders the ADM approach (2.4) not readily applicable. To solve the  $\mathbf{y}$ -subproblem (2.6) efficiently, we propose to linearize the quadratic term  $\frac{\lambda^2}{2} \|B\mathbf{y} - \mathbf{d}\|^2$  and use the linearized subproblem as a surrogate of (2.6). Our reason for linearizing the first term in (2.6) is that the closed-form solution of the linearized subproblem can be derived easily and hence the computation cost of the inner iteration in (2.4) can be reduced.

By linearizing the first quadratic term in (2.6), we have the following problem:

$$\mathbf{y}^{k+1} = \arg \min_{\mathbf{y} \in \Omega} \left\{ \lambda^2 [\langle B^\top (B\mathbf{y}^k - \mathbf{d}), \mathbf{y} - \mathbf{y}^k \rangle + \frac{\tau}{2} \|\mathbf{y} - \mathbf{y}^k\|^2] + \frac{\beta}{2} \|\mathbf{y} - \mathbf{x}^{k+1} + \frac{\mathbf{z}^k}{\beta}\|^2 \right\}, \quad (2.9)$$

where  $\tau$  is the parameter of the proximal term and it should be chosen judiciously (to be specified later). The optimality condition of (2.9) leads to the following variational inequality:

$$(\mathbf{y}' - \mathbf{y}^{k+1})^\top \{ \lambda^2 B^\top (B\mathbf{y}^k - \mathbf{d}) + [\mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})] + \lambda^2 \tau (\mathbf{y}^{k+1} - \mathbf{y}^k) \} \geq 0, \quad \forall \mathbf{y}' \in \Omega. \quad (2.10)$$

from which the solution is given explicitly by:

$$\mathbf{y}^{k+1} = P_\Omega \left\{ \frac{1}{\lambda^2 \tau + \beta} \cdot [-\lambda^2 B^\top (B\mathbf{y}^k - \mathbf{d}) + \lambda^2 \tau \mathbf{y}^k - \mathbf{z}^k + \beta \mathbf{x}^{k+1}] \right\}. \quad (2.11)$$

Note that the computation for obtaining  $\mathbf{y}^{k+1}$  via (2.11) is dominated by the matrix-vector multiplication of  $B^\top B\mathbf{y}^k$ . For image deblurring problems,  $B$  is a discretized differential operator (e.g. the discrete gradient operator). Hence it is sparse and the cost of matrix-vector multiplication is only of  $O(n)$ , see [34].

Combining (2.11) with the formula for  $\mathbf{x}^{k+1}$  in (2.5), we get our linearized ADM method. We note that since closed-form solutions exist for both  $\mathbf{x}$ - and  $\mathbf{y}$ -subproblems, the computation cost at each inner iteration in (2.4) is low ( $O(n \log n)$  for image deblurring problems).

**2.3. Numerical algorithms based on the linearized ADM approach.** In this section, we derive two algorithms for solving (1.1) based on our discussions in §§2.1–2.2. These two different algorithms stem from two different formulations of (1.1). Let  $\rho(\cdot)$  denote the spectral radius. Our first algorithm is based on the formulation (2.1):

**Algorithm 1:** *The first linearized ADM algorithm for (1.1)*

Input  $A, B, \mathbf{c}, \mathbf{d}, \lambda > 0, \beta > 0$  and  $\tau > \rho(B^\top B)$ . Initialize  $(\mathbf{y}, \mathbf{z}) = (\mathbf{y}^0, \mathbf{z}^0)$ ,  $k = 0$ .

**While** not converged, **Do**

1) Generate  $\mathbf{x}^{k+1}$  by solving (2.5):

$$(A^\top A + \beta I)\mathbf{x}^{k+1} = A^\top \mathbf{c} + \mathbf{z}^k + \beta \mathbf{y}^k.$$

2) Generate  $\mathbf{y}^{k+1}$  by (2.11):

$$\mathbf{y}^{k+1} = P_\Omega \left\{ \frac{1}{\lambda^2 \tau + \beta} \cdot [-\lambda^2 B^\top (B\mathbf{y}^k - \mathbf{d}) + \lambda^2 \tau \mathbf{y}^k - \mathbf{z}^k + \beta \mathbf{x}^{k+1}] \right\}.$$

3) Update  $\mathbf{z}^k$  via  $\mathbf{z}^{k+1} = \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$ .

**End Do**

In addition to (2.1), we can reformulate (1.1) into other forms for which ADM is applicable. For instance, (1.1) can be reformulated as:

$$\min_{\mathbf{x} \in \Omega, \mathbf{y} \in \mathbb{R}^n} \left\{ \frac{1}{2} \|A\mathbf{x} - \mathbf{c}\|^2 + \frac{\lambda^2}{2} \|B\mathbf{y} - \mathbf{d}\|^2 : \mathbf{x} - \mathbf{y} = 0 \right\}. \quad (2.12)$$

Then, with the analogous analysis as in §§2.1–2.2, we can easily derive another linearized ADM algorithm based on the reformulation (2.12). Note that the quadratic term  $\frac{1}{2} \|Ax - c\|^2$  should be linearized when (2.12) is considered.

**Algorithm 2:** *The second linearized ADM algorithm for (1.1)*

Input  $A, B, \mathbf{c}, \mathbf{d}$ ,  $\lambda > 0$ ,  $\beta > 0$  and  $\tau > \rho(A^\top A)$ . Initialize  $(\mathbf{y}, \mathbf{z}) = (\mathbf{y}^0, \mathbf{z}^0)$ ,  $k = 0$ .

**While** not converged, **Do**

- 1) Generate  $\mathbf{y}^{k+1}$  by solving  $(\lambda^2 B^\top B + \beta I)\mathbf{y}^{k+1} = \lambda^2 B^\top \mathbf{d} - \mathbf{z}^k + \beta \mathbf{x}^k$ .
- 2) Generate  $\mathbf{x}^{k+1}$  by  $\mathbf{x}^{k+1} = P_\Omega \left\{ \frac{1}{\tau + \beta} [\tau \mathbf{x}^k + \mathbf{z}^k + \beta \mathbf{y}^{k+1} - A^\top (A \mathbf{x}^k - \mathbf{c})] \right\}$ .
- 3) Update  $z^k$  via  $\mathbf{z}^{k+1} = \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$ .

**End Do**

**REMARK 1.** To ensure convergence, we require the proximal parameter  $\tau$  to be sufficiently large in the sense that  $\tau > \rho(B^\top B)$  for Algorithm 1 and  $\tau > \rho(A^\top A)$  for Algorithm 2. For some applications such as image deblurring problems, the estimation of  $\rho(A^\top A)$  and  $\rho(B^\top B)$  is not hard. In fact, when  $A$  is a blurring operator, we usually normalize  $\|A\|_\infty = 1$  to ensure that the blurred image is still within the given dynamic range. Hence  $\rho(A^\top A) = \|A\|^2 \leq \|A\|_\infty \|A\|_1 = 1$  for symmetric blur  $A$ . For discrete gradient operator  $B$ , we have that  $B^\top B$  being the discrete Laplacian operator and hence by the Gershgorin theorem,  $\rho(B^\top B) \leq 8$ , see [34]. Even for generic operators, if we assume periodic boundary condition, then the operators can be diagonalized by Fast Fourier Transform, see e.g. [6, 25], and hence their spectral radius can easily be computed. One can also employ some backtracking techniques to identify an appropriate  $\tau$ , see e.g. [2].

**REMARK 2.** For clearer exposure of the main idea, we fix the value of the penalty parameter  $\beta$  as a positive constant in our proposed algorithms. For strategies adjusting this parameter dynamically, we refer to, e.g. [19].

**3. Convergence.** In this section, we establish the convergence of our proposed algorithms. Because of the similarity, we only prove the convergence of Algorithm 1 and omit the proof for Algorithm 2. Here we assume that a minimizer of (1.1) exists. This is true if  $\ker(A^\top A) \cap \ker(B^\top B) = \emptyset$ . In that case,  $A^\top A + \lambda^2 B^\top B$  is positive definite, and (1.1) is strictly convex and admits a unique solution  $\mathbf{x}^*$  for arbitrarily given  $\mathbf{c}$  and  $\mathbf{d}$ , see e.g. [4]. We note that for image deblurring problems,  $A$  and  $B$  represent a blurring operator and a regularization operator respectively. Since a blurring operator is an integral operator (a low-pass filter) while a regularization operator is a differential operator (a high-pass filter), we generally have  $\ker(A^\top A) \cap \ker(B^\top B) = \emptyset$ .

We first notice that (2.1) is equivalent to finding  $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in \mathbb{R}^n \times \Omega \times \mathbb{R}^n := \Upsilon$  such that

$$\begin{cases} A^\top (Ax^* - c) - z^* = 0, \\ (y' - y^*)^\top (\lambda^2 B^\top (By^* - d) + z^*) \geq 0, \quad \forall y' \in \Omega, \\ x^* - y^* = 0. \end{cases} \quad (3.1)$$

For notational convenience, we denote  $\mathbf{w}^\top = (\mathbf{x}^\top, \mathbf{y}^\top, \mathbf{z}^\top)$  and  $\mathbf{v}^\top = (\mathbf{y}^\top, \mathbf{z}^\top)$ . We let  $\Upsilon^*$  denote the solution set of (3.1). For  $\mathbf{w}^* = (\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in \Upsilon^*$ , we write  $\mathbf{v}^* = (\mathbf{y}^*, \mathbf{z}^*)$ . Since (1.1) admits a solution,  $\Upsilon^*$  is nonempty. In fact,  $\Upsilon^*$  is convex due to the monotonicity of (3.1). Let

$$G = \begin{pmatrix} (\beta + \lambda^2 \tau)I - \lambda^2 B^\top B & 0 \\ 0 & \frac{1}{\beta} I \end{pmatrix}. \quad (3.2)$$

Obviously, for Algorithm 1 with the requirement  $\tau > \rho(B^\top B)$ ,  $G$  is positive definite.

We first present several lemmas which are helpful in establishing the convergence of Algorithm 1.

LEMMA 3.1. *Let*

$$\mathcal{F}(\mathbf{w}) = \begin{pmatrix} A^\top(A\mathbf{x} - \mathbf{c}) - \mathbf{z} \\ \lambda^2 B^\top(B\mathbf{y} - \mathbf{d}) + \mathbf{z} \\ \mathbf{x} - \mathbf{y} \end{pmatrix}.$$

*Then  $\mathcal{F}$  is monotone, i.e.,  $(\mathbf{w}' - \mathbf{w})^\top(\mathcal{F}(\mathbf{w}') - \mathcal{F}(\mathbf{w})) \geq 0$ , for all  $\mathbf{w}'$ ,  $\mathbf{w} \in \Upsilon$ .*

*Proof.* It is elementary, and thus omitted.  $\square$

The following two lemmas present some contractive properties of the sequence generated by Algorithm 1, and they constitute the essential part of our convergence proof.

LEMMA 3.2. *Let  $\{\mathbf{w}^k\}$  be the sequence generated by Algorithm 1. Then, for all  $\mathbf{w}' \in \Upsilon$ , we have*

$$(\mathbf{w}' - \mathbf{w}^{k+1})^\top \left\{ \mathcal{F}(\mathbf{w}^{k+1}) + \begin{pmatrix} \beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ -\beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & (\beta + \lambda^2\tau)I - \lambda^2 B^\top B & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix} (\mathbf{w}^k - \mathbf{w}^{k+1}) \right\} \geq 0. \quad (3.3)$$

*Proof.* Putting (2.5), (2.10) and the fact  $\mathbf{z}^{k+1} = \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$  together, we have the following variational inequalities:

$$\begin{cases} (\mathbf{x}' - \mathbf{x}^{k+1})^\top \{A^\top(A\mathbf{x}^{k+1} - \mathbf{c}) - [\mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^k)]\} \geq 0, \\ (\mathbf{y}' - \mathbf{y}^{k+1})^\top \{\lambda^2 B^\top(B\mathbf{y}^{k+1} - \mathbf{d}) + [\mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})] + \lambda^2\tau(\mathbf{y}^{k+1} - \mathbf{y}^k) + \lambda^2 B^\top B(\mathbf{y}^k - \mathbf{y}^{k+1})\} \geq 0, \\ (\mathbf{z}' - \mathbf{z}^{k+1})^\top \{\mathbf{x}^{k+1} - \mathbf{y}^{k+1} - \frac{1}{\beta}(\mathbf{z}^k - \mathbf{z}^{k+1})\} \geq 0, \end{cases}$$

for all  $\mathbf{w}' \in \lambda$ . Recalling the definition of  $\mathcal{F}$ , the above inequality can be rewritten as

$$(\mathbf{w}' - \mathbf{w}^{k+1})^\top \left\{ \mathcal{F}(\mathbf{w}^{k+1}) + \begin{pmatrix} \beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ -\beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbf{0} \\ (\beta + \lambda^2\tau)(\mathbf{y}^k - \mathbf{y}^{k+1}) - \lambda^2 B^\top B(\mathbf{y}^k - \mathbf{y}^{k+1}) \\ \frac{1}{\beta}(\mathbf{z}^k - \mathbf{z}^{k+1}) \end{pmatrix} \right\} \geq 0, \quad (3.4)$$

for all  $\mathbf{w}' \in \Upsilon$ , and it implies (3.3) immediately.  $\square$

LEMMA 3.3. *Let  $\{\mathbf{v}^k := (\mathbf{y}^k, \mathbf{z}^k)\}$  be generated by Algorithm 1,  $\mathbf{w}^* = (\mathbf{x}^*, \mathbf{v}^*) \in \Upsilon^*$ , and  $G$  be defined in (3.2). Then, we have*

$$(\mathbf{v}^k - \mathbf{v}^*)^\top G(\mathbf{v}^k - \mathbf{v}^{k+1}) \geq \|\mathbf{v}^k - \mathbf{v}^{k+1}\|_G^2 + (\mathbf{z}^k - \mathbf{z}^{k+1})^\top(\mathbf{y}^{k+1} - \mathbf{y}^k). \quad (3.5)$$

*Proof.* By setting  $\mathbf{w}' = \mathbf{w}^*$  in (3.3), we get

$$(\mathbf{w}^* - \mathbf{w}^{k+1})^\top \left\{ F(\mathbf{w}^{k+1}) + \begin{pmatrix} \beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ -\beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ \mathbf{0} \end{pmatrix} - G(\mathbf{v}^k - \mathbf{v}^{k+1}) \right\} \geq 0.$$

Since  $\mathbf{x}^* - \mathbf{y}^* = \mathbf{0}$  and  $\mathbf{x}^{k+1} - \mathbf{y}^{k+1} = \frac{1}{\beta}(\mathbf{z}^k - \mathbf{z}^{k+1})$ , we have

$$\begin{aligned} (\mathbf{v}^{k+1} - \mathbf{v}^*)^\top G(\mathbf{v}^k - \mathbf{v}^{k+1}) &\geq (\mathbf{w}^{k+1} - \mathbf{w}^*)^\top F(\mathbf{w}^{k+1}) + (\mathbf{w}^{k+1} - \mathbf{w}^*)^\top \begin{pmatrix} \beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ -\beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ \mathbf{0} \end{pmatrix} \\ &\geq (\mathbf{w}^{k+1} - \mathbf{w}^*)^\top F(\mathbf{w}^*) + \beta[(\mathbf{x}^{k+1} - \mathbf{x}^*) - (\mathbf{y}^{k+1} - \mathbf{y}^*)]^\top (\mathbf{y}^{k+1} - \mathbf{y}^k) \\ &\geq (\mathbf{z}^k - \mathbf{z}^{k+1})^\top (\mathbf{y}^{k+1} - \mathbf{y}^k). \end{aligned}$$

Combining this with the fact that  $\mathbf{v}^{k+1} - \mathbf{v}^* = \mathbf{v}^k - \mathbf{v}^* - (\mathbf{v}^k - \mathbf{v}^{k+1})$ , we get (3.5).  $\square$

Based on the preceding lemmas, we are able to show some important properties of  $\{\mathbf{v}^k\}$  which ensure the convergence of Algorithm 1 immediately. We summarize these properties in the following lemma.

LEMMA 3.4. *Let  $\{\mathbf{v}^k := (\mathbf{y}^k, \mathbf{z}^k)\}$  be generated by Algorithm 1 and  $\mathbf{w}^* = (\mathbf{x}^*, \mathbf{v}^*) \in \Upsilon^*$ . Then, we have:*

- (i)  $\lim_{k \rightarrow \infty} \|\mathbf{v}^k - \mathbf{v}^{k+1}\|_G = 0$ ;
- (ii)  $\{\mathbf{v}^k\}$  is a bounded sequence;
- (iii)  $\|\mathbf{v}^k - \mathbf{v}^*\|_G^2$  is non-increasing and thus converges.

*Proof.* From  $\mathbf{v}^{k+1} = \mathbf{v}^k - (\mathbf{v}^k - \mathbf{v}^{k+1})$  and (3.5), we get

$$\begin{aligned} &\|\mathbf{v}^k - \mathbf{v}^*\|_G^2 - \|\mathbf{v}^{k+1} - \mathbf{v}^*\|_G^2 \\ &= 2(\mathbf{v}^k - \mathbf{v}^*)^\top G(\mathbf{v}^k - \mathbf{v}^{k+1}) - \|\mathbf{v}^k - \mathbf{v}^{k+1}\|_G^2 \\ &\geq 2\|\mathbf{v}^k - \mathbf{v}^{k+1}\|_G^2 + 2(\mathbf{z}^k - \mathbf{z}^{k+1})^\top (\mathbf{y}^{k+1} - \mathbf{y}^k) - \|\mathbf{v}^k - \mathbf{v}^{k+1}\|_G^2 \\ &= (\beta + \lambda^2 \tau) \|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 - \lambda^2 \|B(\mathbf{y}^k - \mathbf{y}^{k+1})\|^2 + \frac{1}{\beta} \|\mathbf{z}^k - \mathbf{z}^{k+1}\|^2 + 2(\mathbf{z}^k - \mathbf{z}^{k+1})^\top (\mathbf{y}^{k+1} - \mathbf{y}^k). \quad (3.6) \end{aligned}$$

We denote

$$\delta = \frac{\lambda^2(\tau - \rho(B^\top B))}{\beta} > 0 \quad \text{and} \quad \gamma = \frac{\delta + \sqrt{\delta^2 + 4}}{2}.$$

Then, it is easy to show that  $1 < \gamma < 1 + \delta$  and  $1 + \delta - \gamma = 1 - 1/\gamma$ . It follows from the Cauchy-Schwartz inequality that

$$2(\mathbf{z}^k - \mathbf{z}^{k+1})^\top (\mathbf{y}^{k+1} - \mathbf{y}^k) \geq -\gamma \beta \|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 - \frac{1}{\gamma \beta} \|\mathbf{z}^k - \mathbf{z}^{k+1}\|^2.$$

Substituting the above inequality into (3.6), we get

$$\begin{aligned} \|\mathbf{v}^k - \mathbf{v}^*\|_G^2 - \|\mathbf{v}^{k+1} - \mathbf{v}^*\|_G^2 &\geq \beta(1 + \delta - \gamma) \|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 + \left(1 - \frac{1}{\gamma}\right) \frac{1}{\beta} \|\mathbf{z}^k - \mathbf{z}^{k+1}\|^2 \\ &\geq \nu \|\mathbf{v}^k - \mathbf{v}^{k+1}\|_G^2, \end{aligned} \quad (3.7)$$

where

$$\nu := \min \left\{ \frac{\beta(1 + \delta - \eta)}{\beta + \tau}, 1 - \frac{1}{\gamma} \right\} = \frac{\beta}{\beta + \tau} \left(1 - \frac{1}{\gamma}\right) > 0.$$

From that, it is trivial to derive the statements of the lemma based on (3.7).  $\square$

Now, we are ready to present the convergence of Algorithm 1 in the following theorem.

THEOREM 3.5. *The sequence  $\{\mathbf{v}^k := (\mathbf{y}^k, \mathbf{z}^k)\}$  generated by Algorithm 1 converges to  $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ , where  $\{(\mathbf{x}^*, \mathbf{y}^*)\}$  is a solution of (2.1).*

*Proof.* It follows from (i) of Lemma 3.4 that

$$\lim_{k \rightarrow \infty} \|\mathbf{y}^k - \mathbf{y}^{k+1}\| = 0, \quad \lim_{k \rightarrow \infty} \|\mathbf{z}^k - \mathbf{z}^{k+1}\| = 0. \quad (3.8)$$

From (ii) of Lemma 3.4,  $\{\mathbf{v}^k\}$  has at least one accumulation point and we denote it by  $\mathbf{v}^\infty = (\mathbf{y}^\infty, \mathbf{z}^\infty)$ . There also exists a subsequence  $\{\mathbf{v}^{k_j}\}$  converging to  $\mathbf{v}^\infty$ , i.e.,  $\mathbf{y}^{k_j} \rightarrow \mathbf{y}^\infty$  and  $\mathbf{z}^{k_j} \rightarrow \mathbf{z}^\infty$ . Furthermore, from (2.5), we get

$$x^{k+1} = (A^\top A + \beta I)^{-1}(A^\top \mathbf{c} + \beta \mathbf{y}^k + \mathbf{z}^k).$$

Since  $\mathbf{y}^{k_j} \rightarrow \mathbf{y}^\infty$  and  $\mathbf{z}^{k_j} \rightarrow \mathbf{z}^\infty$ , we have  $\mathbf{y}^{k_j} - \mathbf{y}^{k_j-1} \rightarrow \mathbf{0}$  and  $\mathbf{z}^{k_j} - \mathbf{z}^{k_j-1} \rightarrow \mathbf{0}$ . Thus, we conclude that

$$x^{k_j} \rightarrow x^\infty := (A^\top A + \beta I)^{-1}(A^\top \mathbf{c} + \beta \mathbf{y}^\infty + \mathbf{z}^\infty). \quad (3.9)$$

Therefore,  $(x^\infty, y^\infty, z^\infty)$  is also an accumulation point of the sequence  $\{w^k = (x^k, y^k, z^k)\}$ .

Next, we show that  $(\mathbf{x}^\infty, \mathbf{y}^\infty, \mathbf{z}^\infty)$  satisfies the optimality condition for (2.1). First, substituting (3.8) into (3.4), we have

$$\lim_{k \rightarrow \infty} (\mathbf{w} - \mathbf{w}^{k+1})^\top F(\mathbf{w}^{k+1}) \geq 0, \quad \forall \mathbf{w} \in \Upsilon. \quad (3.10)$$

Thus, any accumulation point of  $\{\mathbf{w}^k\}$  is a solution point of (3.1). So,  $(\mathbf{x}^\infty, \mathbf{y}^\infty, \mathbf{z}^\infty)$  is a solution point of (3.1). Since the inequality (3.7) is true for all solution points of (3.1), we have that

$$\|\mathbf{v}^{k+1} - \mathbf{v}^\infty\| \leq \|\mathbf{v}^k - \mathbf{v}^\infty\|, \quad \forall k \geq 0. \quad (3.11)$$

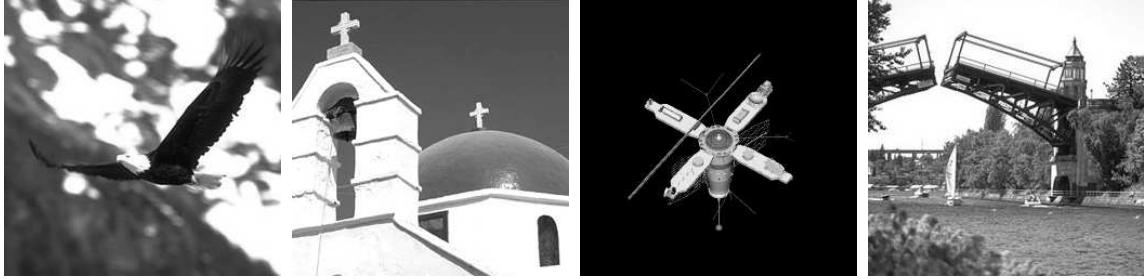
Thus,  $\mathbf{v}^\infty$  is the only accumulation point of the sequence  $\{\mathbf{v}^k\}$ , and so  $\{\mathbf{v}^k\}$  converges to  $\mathbf{v}^\infty$ . Accordingly,  $\{\mathbf{w}^k\}$  converges to  $(\mathbf{x}^\infty, \mathbf{y}^\infty, \mathbf{z}^\infty)$  which is a solution of (3.1) and therefore also of (2.1).  $\square$

**4. Numerical experiments.** In this section, we test the proposed linearized ADM algorithms on image deblurring problems

$$\min_{\mathbf{x} \leq \mathbf{u}} \left\{ \frac{1}{2} \|A\mathbf{x} - \mathbf{c}\|^2 + \frac{\lambda^2}{2} \|B\mathbf{x}\|^2 \right\}, \quad (4.1)$$

where  $A$  is a blurring operator,  $B$  is a regularization operator and  $\mathbf{c}$  is the observed image. Obviously, (4.1) is a special case of (1.1) with  $\mathbf{d} = 0$ . We compare the proposed Algorithms 1 and 2 with the reduced Newton method in [23] (denoted by RN) and the affine scaling method in [7] (denoted by AS). It has been shown numerically that RN and AS are better than the projection-type methods and some existing Newton-like methods. All the codes were written with MATLAB 7.8 (R2009a) and were run on a T6500 notebook with the Intel Core 2 Duo CPU at 2.1 GHz and 2GB of memory.

For the purpose of comparison, we test the same  $256 \times 256$  images as in [23], i.e., the Eagle, Church, Satellite and Bridge images, see Figure 4.1. Accordingly,  $m = n = 65,536$  in model (1.1) for these images. As in [23], the blurring matrix  $A$  is chosen to be the out-of-focus blur and the matrix  $B$  is taken to be the gradient matrix. Under the periodic boundary conditions for  $\mathbf{x}$ , both  $B^\top B$  and  $A^\top A$  are block circulant matrices with circulant blocks. Thus they are diagonalizable by the 2D discrete Fourier transforms, see [6], and hence our Algorithms 1 and 2 are of  $O(n \log n)$  operations per iteration. The observed image  $\mathbf{c}$  is expressed as  $\mathbf{c} = A\bar{\mathbf{x}} + \eta\mathbf{r}$ , where  $\bar{\mathbf{x}}$  is the true image,  $\mathbf{r}$  is a random vector with entries distributed as standard normal, and  $\eta$  is the level of noise. The bound constraints are set to  $l_i = 0$  and  $u_i = 255$  for all  $i = 1, \dots, n$ . We employ the MATLAB scripts:  $\mathbf{A} = \text{fspecial('average', alpha)}$  and  $\mathbf{C} = \text{imfilter(X, A, 'circular', 'conv')} +$

FIG. 4.1. *Original images*

`eta*randn(m,n)` to produce the blurred images corrupted by the averaging kernel of different sizes. Here, `alpha` is the size of the kernel, `X` denotes the original image, and `C` represents the observed image.

We stop Algorithms 1 and 2 when the relative change  $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|/\|\mathbf{x}^{k-1}\| \leq 10^{-4}$ . We measured the quality of restoration by the peak signal-to-noise ratio (PSNR), which is measured in decible (dB) and is defined by:

$$\text{PSNR}(\mathbf{x}) = 20 \log_{10} \frac{\mathbf{x}_{\max}}{\sqrt{\text{Var}(\mathbf{x}, \bar{\mathbf{x}})}} \quad \text{with} \quad \text{Var}(\mathbf{x}, \bar{\mathbf{x}}) = \frac{\sum_{j=0}^{n-1} [\bar{\mathbf{x}}(j) - \mathbf{x}(j)]^2}{n},$$

where  $\bar{\mathbf{x}}$  is the original image and  $\mathbf{x}_{\max}$  is the maximum possible pixel value of the image  $\mathbf{x}$ .

In our experiments, we set  $\beta = 0.1$  for Algorithms 1 and 2. In our proof of convergence, we show that the convergence of Algorithm 1 (Algorithm 2, respectively) is ensured when  $\tau > \rho(B^\top B)$  ( $\tau > \rho(A^\top A)$ , respectively). Hence we set  $\tau = 1.05 \cdot \rho(B^\top B)$  ( $\tau = 1.05 \cdot \rho(A^\top A)$ , respectively) in the implementation of Algorithm 1 (Algorithm 2, respectively). Note that since  $B^\top B$  and  $A^\top A$  can be diagonalized by Fast Fourier Transform, their spectral radius can easily be obtained.

In the first set of experiments, we set  $\lambda = 0.1$  as in [23]. In Table 4.1, we report the performance of RN, AS, Algorithm 1 (Alg1) and Algorithm 2 (Alg2) for different levels of noise  $\eta$  and blur size `alpha`. We report the computing time in seconds (denoted by “Time(s)”), PSNR of the recovered images (denoted by “PSNR(dB)”) and the function value of the model (4.1) (denoted by “Objfn-end”) when the iteration is terminated. Note that the Newton-type methods RN and AS both require an interior point inside  $[0, 255]^n$  as the initial iterate. We here apply the method as in [23] to generate their initial iterate, i.e., we solve the unconstrained version of (4.1):

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \frac{1}{2} \|\mathbf{Ax} - \mathbf{c}\|^2 + \frac{\lambda^2}{2} \|\mathbf{Bx}\|^2 \right\}, \quad (4.2)$$

and project the solution onto the box  $[1, 254]^n$  to obtain the initial iterate for RN and AS. Note that solving (4.2) amounts to solving its normal equation  $(A^\top A + \lambda^2 B^\top B)\mathbf{x} = A^\top \mathbf{c}$ . On the contrary, the proposed Algorithms 1 and 2 can use any arbitrary image as the initial iterate. In our experiment we simply start the iterations of Algorithms 1 and 2 from the blurred image  $\mathbf{c}$ . In the last column of Table 4.1, the “PJ” column, we also report the PSNR of the projected solution of (4.2). More precisely, we solve (4.2) by the least squares method, and then we project the solution onto the box  $[0, 255]^n$ . For each test case, we repeat the experiment three times and report the average performance.

The efficiency of our Algorithms 1 and 2 are shown clearly in Table 4.1 where the best results are marked by boldface. First we emphasize that the PSNR obtained by any of the four algorithms for the constrained model (4.1) is higher than that obtained by the projected solution of the unconstrained model (4.2) by 0.2

TABLE 4.1  
*Numerical comparison of RN, AS, Algorithm 1 and Algorithm 2*

Image alpha	$\eta$	Time (s)				PSNR (dB)				Objfn-end ( $\times 10^5$ )				PJ (dB)
		RN	AS	Alg1	Alg2	RN	AS	Alg1	Alg2	RN	AS	Alg1	Alg2	
Eagle	3	4.49	1.44	<b>0.52</b>	1.70	32.80	32.78	32.83	<b>32.96</b>	2.885	2.885	<b>2.879</b>	2.886	32.26
	5	5.24	1.53	<b>0.58</b>	1.98	29.36	29.37	29.37	<b>29.50</b>	6.312	6.311	<b>6.301</b>	6.313	28.87
	7	5.41	1.77	<b>0.61</b>	2.08	26.83	26.84	26.84	<b>26.94</b>	11.32	11.32	<b>11.30</b>	11.32	26.38
Church	3	5.28	1.78	<b>0.48</b>	1.69	30.92	30.90	<b>30.94</b>	<b>30.94</b>	3.895	3.894	<b>3.887</b>	3.896	30.08
	5	6.19	1.60	<b>0.58</b>	1.86	28.57	28.56	28.58	<b>28.66</b>	7.442	7.442	<b>7.425</b>	7.443	27.83
	7	6.21	1.67	<b>0.63</b>	2.17	26.51	26.51	26.52	<b>26.60</b>	12.57	12.57	<b>12.54</b>	12.57	25.82
Satellite	3	6.15	2.29	<b>0.97</b>	3.59	<b>28.97</b>	28.88	<b>28.97</b>	28.93	3.463	3.464	<b>3.461</b>	3.463	27.91
	5	15.76	2.27	<b>0.92</b>	3.89	28.53	28.47	<b>28.54</b>	28.52	8.369	8.370	<b>8.365</b>	8.368	26.86
	7	11.24	2.46	<b>1.03</b>	4.03	27.90	27.87	<b>27.91</b>	<b>27.91</b>	15.72	15.72	<b>15.71</b>	<b>15.71</b>	25.65
Bridge	3	6.08	2.39	<b>0.88</b>	2.94	<b>24.38</b>	24.33	<b>24.38</b>	24.26	4.588	4.589	<b>4.587</b>	4.592	24.06
	5	6.11	2.19	<b>0.89</b>	3.31	<b>23.71</b>	23.66	<b>23.71</b>	23.67	8.831	8.833	<b>8.830</b>	8.835	23.34
	7	10.09	2.39	<b>0.98</b>	3.78	22.90	22.88	22.91	<b>22.92</b>	15.25	15.25	<b>15.24</b>	15.25	22.50

TABLE 4.2  
*Numerical comparison of Algorithm 1 and Algorithm 2 with best  $\lambda^*$*

Image alpha	$\eta$	$\lambda^*$		PSNR (dB)		Time (s)		$\lambda^*$	PJ (dB)
		Alg1	Alg2	Alg1	Alg2	Alg1	Alg2		
Eagle	3	0.16	0.16	37.65	37.64	0.47	0.92	0.17	33.20
	5	0.23	0.23	32.03	32.03	0.50	0.77	0.24	31.80
	7	0.30	0.30	31.02	31.02	0.63	0.81	0.32	30.86
Church	3	0.10	0.10	30.96	30.96	0.42	1.70	0.11	30.11
	5	0.15	0.15	29.25	29.25	0.44	1.17	0.16	28.69
	7	0.19	0.19	28.18	28.18	0.53	0.91	0.21	27.80
Satellite	3	0.05	0.04	29.66	29.66	1.56	9.27	0.10	27.92
	5	0.08	0.08	28.52	28.51	1.02	5.22	0.15	27.26
	7	0.12	0.12	27.84	27.83	0.86	3.31	0.21	26.85
Bridge	3	0.06	0.06	24.96	24.88	1.22	5.13	0.06	24.48
	5	0.09	0.09	23.73	23.71	0.89	3.83	0.10	23.33
	7	0.12	0.12	22.97	22.96	0.77	2.95	0.13	22.64

to 2.2 dB (see the ‘PJ’ column in the table). Thus, it pays to solve the constrained model (4.1). Secondly, we see that Algorithm 1 gives the smallest objective function values amongst all four methods and is the fastest one amongst them. We note that Algorithm 1 involves matrix-vector multiplication of  $B^\top B$  which can be done in  $O(n)$  whereas Algorithm 2 involves matrix-vector multiplication of  $A^\top A$  which is done in  $O(n \log n)$ . Thus the cost of iteration is cheaper for Algorithm 1.

In terms of PSNR, Algorithm 1 is either the best one or within 0.15 dB from the best one. This difference is actually due to the fact that we here fixed  $\lambda = 0.1$  for all cases. The regularization parameter  $\lambda$  should be chosen according to the noise level. In Table 4.2, we repeat the experiments for Algorithms 1 and 2 with the best  $\lambda$  chosen by trial and error. We see from the table that with the best  $\lambda$ , Algorithms 1 can actually attain slightly better PSNR than Algorithm 2 does. Again, Algorithm 1 is much faster than Algorithm 2. Note also that the PSNR of the constrained model can be 4.4dB higher than that of the projected solution.

To further verify the efficiency and robustness of our proposed algorithms, we use the MATLAB script `fspecial` to generate four more different blurred images and compare their restoration results. We add the ‘average’ blur to the satellite image with  $\text{alpha} = 3$  and noise level  $\eta = 4$ , and to the church image with  $\text{alpha} = 5$  and  $\eta = 2$ . For the eagle image, we adopted the ‘motion’ blur kernel with  $\text{len} = 5$ , and  $\text{theta} = 15$  and noise level  $\eta = 2$ . For the bridge image, we applied the ‘Gaussian’ blur kernel with  $\text{hsize} = 5$ ,  $\sigma = 4$  and noise level  $\eta = 4$ . The restored images are shown in Figure 4.2 with timing and accuracy given. Again we see that our Algorithm 1 outperforms other methods in terms of timing and the PSNR is either the best one or differs from the best one by 0.01dB.

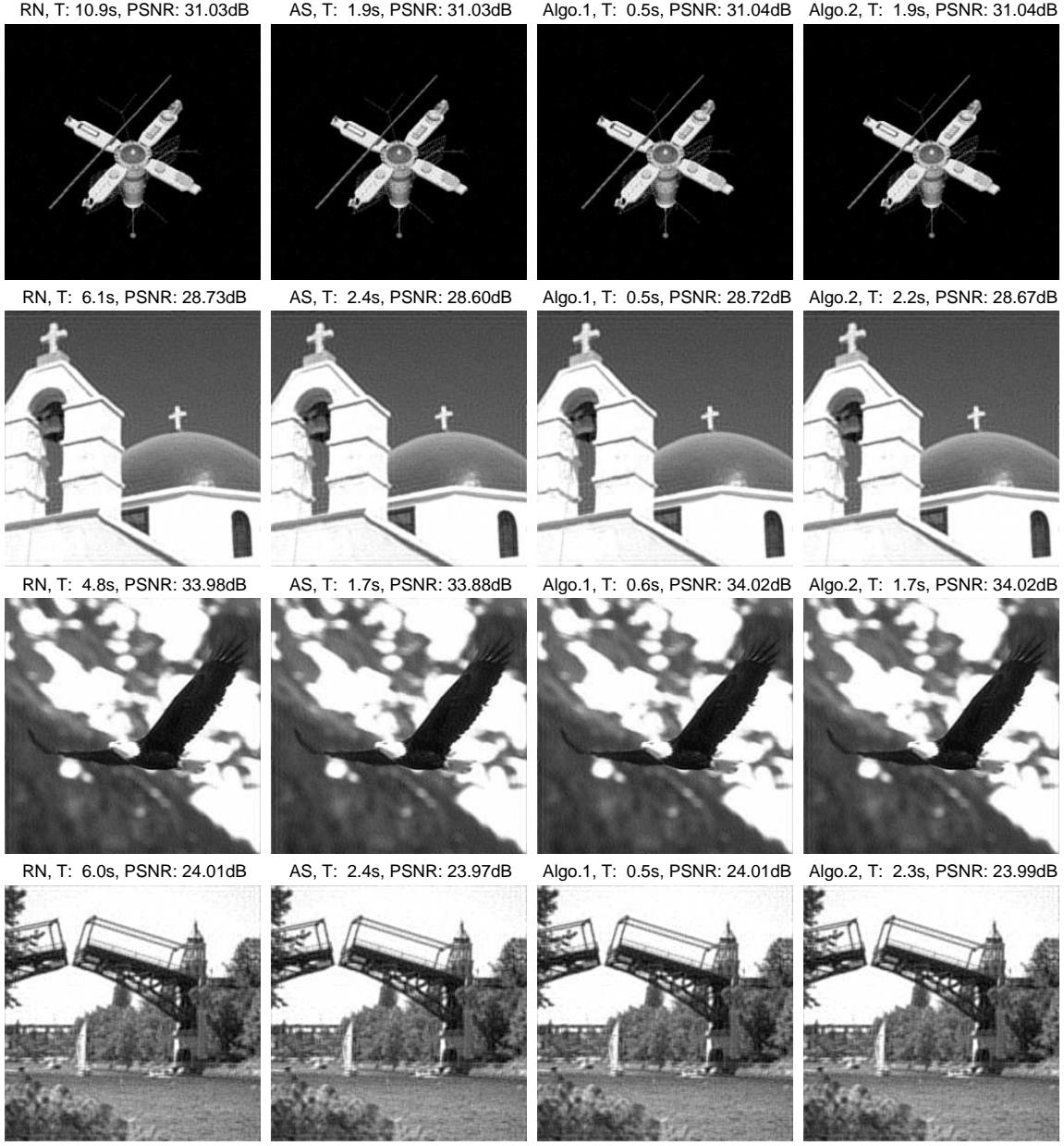


FIG. 4.2. From left to right: Results recovered by RN, AS, Algos. 1 and 2, respectively. “T” represents running time.

**5. Concluding remarks.** This paper provides a novel approach for solving constrained linear least-squares problems by combining linearization techniques with the alternating direction method (ADM). The approach belongs to the category of inexact ADMs where inexact solutions of ADM subproblems are obtained via linearizing some quadratic terms involved. Since ADM is a first-order method which in general requires more outer iterations compared to second-order type methods, a major concern of making ADM efficient is to alleviate the difficulty of obtaining approximate solutions in the inner iterations. The proposed linearization technique realizes this philosophy well, as all the linearized inner subproblems have closed-form solutions.

We have proved the convergence of the proposed linearized ADM approach and applied it to solve some image deblurring problems. Comparing to Newton-type methods, the linearized ADM approach is more

efficient both in terms of speed and the quality of the restored images. Moreover, the linearized ADM approach does not require a good initial guess to start its iteration, while Newton-type methods are usually sensitive to the choice of the initial iterant.

One extension of this research is to develop inexact ADMs for more complex problems in image processing, such as the multiframe deblurring problem where the objective function takes the form

$$\sum_{i=1}^K \frac{1}{2} \|A_i \mathbf{x} - \mathbf{c}_i\|^2 + \frac{\lambda^2}{2} \|B\mathbf{x}\|^2.$$

#### REFERENCES

- [1] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*. IMA J. Num. Analysis, 8, pp. 141–148, 1988.
- [2] A. Beck and M. Teboulle, *A fast iterative shrinkage-Thresholding algorithms for inner inverse problems*, SIAM. J. Imaging Sciences, 2, pp. 183–202, 2009.
- [3] S. Bellavia, M. Macconi and B. Morini, *An interior Newton-like method for nonnegative least-squares problems with degenerate solution*, Numerical Linear Algebra with Applications, 13, pp. 825–846, 2006.
- [4] A. Björck, Numerical Methods for Least Squares Problems, SIAM, 1996.
- [5] D. Calvetti, G. Landi, L. Reichel and F. Sgallari, *Nonnegativity and iterative methods for ill-posed problems*, Inverse Problems 20, pp. 1747–1758, 2004.
- [6] R. H. Chan and M. K. Ng, *Conjugate gradient method for Toeplitz systems*, SIAM Review, 38, pp. 427–482, 1996.
- [7] R. H. Chan, B. Morini and M. Porcelli, *Affine scaling methods for image deblurring problems*, American Institute of Physics Conference Proceedings 1281, vol.2, pp. 1043–1046, 2010.
- [8] R. H. Chan, J. F. Yang and X. M. Yuan, *Alternating direction method for image inpainting in wavelet domain*, submitted, 2010.
- [9] T. Coleman and Y. Li, *An interior trust-region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6, 418–445, 1996.
- [10] M. Ehrgott, I. Winz, *Interactive decision support in radiation therapy treatment planning*, OR Spectrum 30, pp. 311–329, 2008.
- [11] E. Esser, *Applications of Lagrangian-Based alternating direction methods and connections to split Bregman*, UCLA CAM Report 09-31, 2009.
- [12] M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming problems*, Comput. Optim. Appl., 1, pp.93–111, 1992.
- [13] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2, 17–40, 1976.
- [14] D. Gabay, *Applications of the method of multipliers to variational inequalities*, Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems, North Holland, Amsterdam, Holland, 1983, pp. 299–331.
- [15] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.
- [16] T. Goldstein and S. Osher, *The split Bregman method for  $L_1$ -Regularized Problems*, SIAM J. Imag. Sci., 2, pp. 323–343, 2009.
- [17] R. Gonzalez, R. Woods, Digital Image Processing, 3rd ed., Prentice Hall, NJ, 2008.
- [18] B. S. He, L. Z. Liao, D. Han and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Math. Program., 92, pp. 103–118, 2002.
- [19] B. S. He, H. Yang and S. L. Wang, *Alternating direction method with self-adaptive penalty parameters for monotone variational inequality*, J. Optim. Theory Appli., 106, pp. 337–356, 2000.
- [20] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appli., 4 , pp. 303–320, 1969.
- [21] Y. M. Huang, M. K. Ng and Y. W. Wen, *A fast total variational minimization method for image restoration*, SIAM . Multi. Modeling Simul., 7, pp. 774–795, 2008.
- [22] C. L. Lawson, and R. J. Hanson, Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [23] B. Morini, M. Porcelli and R. H. Chan, *A reduced Newton method for constrained linear least-squares problems*, J. Comput. Applied Math., 233, pp. 2200–2212, 2010.
- [24] S. Morigi, L. Reichel, F. Sgallari and F. Zama, *An iterative method for linear discrete ill-posed problems with box constraints*, J. Comput. Applied Math., 198, pp. 505–520, 2007.

- [25] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21, pp. 851–866, 1999.
- [26] M. K. Ng, F. Wang and X. M. Yuan, *Fast minimization methods for solving constrained total-variation superresolution image reconstruction*, Multi. Syst. Signal Proces., to appear.
- [27] M. K. Ng, P. A. Weiss and X. M. Yuan, *Solving constrained total-variation problems via alternating direction methods*, SIAM J. Sci. Comput., 32(5), pp. 2710–2736, 2010.
- [28] L. F. Portugal, J. J. Joaquim and L. N. Vicente, *A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables*, Math. Comp. 63, pp. 625–643, 1994.
- [29] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, In Optimization edited by R. Fletcher, pp. 283–298, Academic Press, New York, 1969.
- [30] M. Rojas and T. Steihaug, *An interior-point trust-region-based method for large-scale non-negative regularization*, Inverse Problems, 18, pp. 1291–1307, 2002.
- [31] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60, pp. 259–268, 1992.
- [32] S. Setzer, *Split Bregman algorithm, Douglas-Rachford splitting, and frame shrinkage*, Lect. Notes Comput. Sci. 5567, pp. 464–476, 2009.
- [33] S. Setzer, G. Steidl and T. Tebuber, *Deblurring Poissonian images by split Bregman techniques*, J. Visual Commun. Image Repres. , 21, pp. 193–199, 2010.
- [34] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Texts in Applied Mathematics, No. 12, Springer, 2007.
- [35] M. Tao and X. M. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM Journal on Optimization, 21(1), pp. 57–81, 2011.
- [36] A. Tikhonov, and V. Arsenin, Solution of Ill-Posed problems, Winston, Washington, DC, 1977.
- [37] P. Tseng, *Applications of Splitting Algorithm to Decomposition in Convex Programming and Variational Inequalities*, SIAM Journal on Control and Optimization, 29 (1991), pp. 119–138.
- [38] J. F. Yang and Y. Zhang, *Alternating direction algorithms for  $l_1$ -problems in compressive sensing*, TR09-37, CAAM, Rice University.
- [39] J. F., Yang, Y. Zhang, and W. Yin, *A fast alternating direction method for  $TVL1-L2$  signal reconstruction from partial Fourier data*, IEEE Trans. Selected Topics in Signal Processing, to appear.