East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.270812.161112a

Vol. 2, No. 4, pp. 326-341
November 2012

Linearized Alternating Direction Method of Multipliers for Constrained Linear Least-Squares Problem

Raymond H. Chan[1,*], Min Tao[2] and Xiaoming Yuan[3]

[1] *Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China.*
[2] *School of Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China.*
[3] *Department of Mathematics, Hong Kong Baptist University, Hong Kong, China.*

Received 27 August 2012; Accepted (in revised version) 16 November 2012

Available online 29 November 2012

Abstract. The alternating direction method of multipliers (ADMM) is applied to a constrained linear least-squares problem, where the objective function is a sum of two least-squares terms and there are box constraints. The original problem is decomposed into two easier least-squares subproblems at each iteration, and to speed up the inner iteration we linearize the relevant subproblem whenever it has no known closed-form solution. We prove the convergence of the resulting algorithm, and apply it to solve some image deblurring problems. Its efficiency is demonstrated, in comparison with Newton-type methods.

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

Key words: Linear least-squares problems, alternating direction method of multipliers, linearization, image processing.

1. Introduction

In this paper, we consider the constrained linear least-squares problem

$$\min_{\mathbf{l}\le\mathbf{x}\le\mathbf{u}} \left\{ \frac{1}{2}\|A\mathbf{x}-\mathbf{c}\|^2 + \frac{\lambda^2}{2}\|B\mathbf{x}-\mathbf{d}\|^2 \right\}, \tag{1.1}$$

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

[*] Corresponding author. *Email addresses:* rchan@math.cuhk.edu.hk (R. H. Chan), taom@njupt.edu.cn (M. Tao), xmyuan@hkbu.edu.hk (X. Yuan)

Clearly, the problem (1.1) can be written as $\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} ||C\mathbf{x}-\mathbf{e}||^2/2$ for some $C \in \mathbb{R}^{2m\times n}$ and $\mathbf{e} \in \mathbb{R}^{2m}$, and there are standard solution procedures such as the Newton-type or interior point methods [9,23,29]. However, our emphasis here is on applications where A and B are two different types of operator, and the design of better algorithms to capture their properties — e.g. 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) becomes the Tikhonov regularization with $\mathbf{c}$ the observed image, λ^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 [37]. In our numerical experiments discussed in Section 4, by imposing the box constraints we find that the peak signal-to-noise ratio of the restored images can be increased by 0.2 to 2.2 decibles, so it pays to solve the constrained problem. In addition, (1.1) also serves as one of the two subproblems for the splitting algorithm in Ref. [22], which solves the deblurring problem with total-variational regularization where $B = I$ (the identity matrix) and $\mathbf{d}$ is an approximation of $\mathbf{x}$ [32]. Other applications of (1.1) include contact problems, control problems, and intensity-modulated radiotherapy problems [4,10].

In this paper, we develop fast solvers for the problem (1.1) that exploit the properties of A and B. In the literature, algorithms for solving (1.1) are essentially Newton-type methods in the context of interior point methods — e.g. some that have been proposed to solve the nonnegative least-squares problem [9,29]. Interior point methods are particularly tailored to ill-posed problems arising in image reconstruction [5,31]. By formulating the Karush-Kuhn-Tucker conditions as a system of nonlinear equations, a Newton-type method was proposed to solve the nonnegative least-squares problem [3]. A precondition technique was applied 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. This work inspired the reduced Newton method presented in Ref. [24], to solve a subsystem of the inner linear system corresponding only to components of the iterate that are not close to the boundary. Since this subsystem is smaller and less ill-conditioned compared to the subsystem in Ref. [3], the reduced Newton method outperforms projection-type methods and some interior point Newton methods for image deblurring problems. A quite recent approach is the affine scaling method [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].

Here we apply the alternating direction method of multipliers (ADMM) originally proposed in Refs. [13,15] to the problem (1.1), where we note the objective function is a sum of two least-squares terms linked together by a common variable $\mathbf{x}$. By introducing an auxiliary variable, we separate the two least-squares terms and apply the ADMM directly. The ADMM decomposes (1.1) into two easier subproblems, each of which is a least-squares problem with only one quadratic term in the objective function. However, if neither of the matrices A or B in (1.1) is the identity matrix so the respective ADMM subproblem does not have a closed-form solution, it is usually difficult to solve. In this paper, we apply a linearization such that a closed-form solution to the resulting linearized ADMM subproblem can be derived readily. For image deblurring problems, the cost per ADMM iteration is about O($n \log n$), where n is the number of pixels in the image. We prove that our linearized ADMM method converges to the minimizer of (1.1), and then show the efficiency of the method by testing it on some image deblurring problems.

The rest of this paper is organized as follows. In Section 2, we present the linearized ADMM approach to solve (1.1), and in Section 3 we establish the convergence of the derived algorithms. Our numerical tests of the linearized ADMM, in comparison with some existing Newton-type methods, are reported in Section 4. Finally, some conclusions are reached in Section 5.

2. The Linearized ADMM Approach

In this Section, we present the linearized ADMM approach and derive the algorithms for solving (1.1). The problem is first reformulated so that the ADMM can be applied, and we then consider the ADMM subproblem linearization where the closed-form solution is readily derived.

2.1. The ADMM approach

To some extent, the ADMM is an improved variant of the classical augmented Lagrangian method (ALM) [21, 30] for solving linearly constrained convex programming problems [21, 30] when the objective functions are in a separable form — i.e. a sum of individual functions without crossed variables. The efficiency of the ADMM has been well documented in the context of convex programming and variational inequalities — e.g. see Refs. [12, 16, 19, 38]. Recently, the ADMM has been used to successfully address many applications in the area of image processing [8, 11, 27, 28, 33, 34, 36], and the relationship between the ADMM and the split Bregman method [17] has been discussed [11, 33, 34].

In order to apply the ADMM, we first reformulate (1.1) as

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

where $\Omega := \{\mathbf{l} \le \mathbf{x} \le \mathbf{u} \mid \mathbf{x} \in \mathbb{R}^n\}$ and $\mathbf{y}$ is an artificial variable. Strictly speaking, (2.1) is a typical linearly-constrained convex programming problem, so the ALM is readily applicable. Specifically, from the augmented Lagrangian function of (2.1)

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

where $\mathbf{z} \in \mathbb{R}^n$ is the Lagrange multiplier and $\beta > 0$ is a penalty parameter for violation of the linear constraints, the ALM iterative scheme for (2.1) to render $(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}, \mathbf{z}^{k+1})$ from $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{z}^k)$ at each step is

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

It is notable that the direct application of the ALM (2.3) treats (2.1) as a generic linearly constrained convex programming problem without any consideration of its separable structure, such that the variables $\mathbf{x}^{k+1}$ and $\mathbf{y}^{k+1}$ must be solved simultaneously. In contrast, the ADMM decomposes the minimization in (2.3) into two subproblems to solve for $\mathbf{x}^{k+1}$ and $\mathbf{y}^{k+1}$ consecutively, under the scheme

$$\begin{cases} \mathbf{x}^{k+1} \leftarrow \arg\min\limits_{\mathbf{x}\in\mathbb{R}^n} \mathscr{L}_{\mathscr{A}}(\mathbf{x},\mathbf{y}^k,\mathbf{z}^k)\,, \\ \mathbf{y}^{k+1} \leftarrow \arg\min\limits_{\mathbf{y}\in\Omega} \mathscr{L}_{\mathscr{A}}(\mathbf{x}^{k+1},\mathbf{y},\mathbf{z}^k)\,, \\ \mathbf{z}^{k+1} \leftarrow \mathbf{z}^k - \beta(\mathbf{x}^{k+1}-\mathbf{y}^{k+1})\,. \end{cases} \tag{2.4}$$

The ADMM therefore belongs to the class of splitting methods due to this decomposition feature, which enables separable structures in the objective function of (2.1) to be exploited, as in the $\mathbf{x}$-subproblem in (2.4) with the closed-form solution discussed below. Let us now proceed to consider the solution of the two subproblems in (2.4), where our linearizing idea is introduced.

The $\mathbf{x}$-subproblem in (2.4) is a least-squares problem without constraints, corresponding to the normal equation

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

For image deblurring problems, A is a spatially-invariant blurring operator. Under periodic boundary conditions for $\mathbf{x}$, $A^\top A$ becomes a block circulant matrix with circulant blocks [6], so it can be diagonalized by the two-dimensional fast Fourier transform (FFT). Thus (2.5) can be solved exactly by using two FFTs and taking one inverse FFT [6,26], so the computational cost is $O(n\log n)$. It is easy to show that the $\mathbf{y}$-subproblem in (2.4) is equivalent to

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

with solution implicitly given by the projection equation

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

where P_Ω denotes the projection operator onto Ω under the Euclidean norm. For deblurring problems with Tikhonov regularization (e.g. see [18]), we have $B = I$ in (2.1) so the explicit closed-form solution of (2.7) is

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

On the other hand, if $B\neq I$ a closed-form solution of (2.7) is generally unavailable, so we propose a linearization to obtain a useful approximate solution for $\mathbf{y}$.

2.2. The linearized ADMM approach for (2.1)

When there is no known closed-form solution to (2.6), we propose to treat the $\mathbf{y}$-subproblem efficiently by linearizing the quadratic term $\lambda^2\|B\mathbf{y}-\mathbf{d}\|^2/2$ and adopting the linearized subproblem as a surrogate for (2.6). The closed-form solution of the linearized subproblem can easily be derived, so the computational cost of the inner iteration in (2.4) is reduced. On linearizing the first quadratic term in (2.6), the subproblem reduces to

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

where the parameter τ of the proximal term will be chosen judiciously later. The optimality condition of (2.9) leads to the variational inequality

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

yielding the explicit solution

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

The computation to obtain $\mathbf{y}^{k+1}$ is dominated by the matrix-vector multiplication in $B^\top B\mathbf{y}^k$. However, for image deblurring problems B is a discretized differential operator (e.g. the discrete gradient operator) that is sparse, so the cost of the matrix-vector multiplication is only $O(n)$ — cf. [35].

The combination of (2.5) with (2.11) constitutes our linearized ADMM method, where closed-form solutions exist for not only the $\mathbf{x}$-subproblem but also the $\mathbf{y}$-subproblem, such that the computational cost at the inner iterations in (2.4) is low — e.g. $O(n\log n)$ for image deblurring problems.

2.3. Numerical algorithms based on the linearized ADMM approach

Based on our discussion in Sections 2.1 and 2.2, we now derive two different algorithms for solving (1.1), which stem from two different formulations of the problem.

Let $\rho(\cdot)$ denote the spectral radius. Our first algorithm is based on formulation (2.1):

```
Algorithm 1: The first linearized ADMM 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):
           $\mathbf{y}^{k+1} = P_\Omega\left\{\frac{1}{\lambda^2\tau+\beta}\cdot\left[-\lambda^2B^\top(B\mathbf{y}^k-\mathbf{d})+\lambda^2\tau\mathbf{y}^k-\mathbf{z}^k+\beta\mathbf{x}^{k+1}\right]\right\}$.
        3) Update $\mathbf{z}^k$ via  $\mathbf{z}^{k+1} = \mathbf{z}^k-\beta(\mathbf{x}^{k+1}-\mathbf{y}^{k+1})$.

    End Do
```

However, we can reformulate (1.1) in another way, where the ADMM is also applicable — e.g. (1.1) can be reformulated as

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

so on linearizing the quadratic term $\|Ax-c\|^2/2$ in analysis similar to that in Sections 2.1 and 2.2 we readily obtain another linearized ADMM algorithm:

```
Algorithm 2: The second linearized ADMM 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
           $\mathbf{x}^{k+1} = P_\Omega\left\{\frac{1}{\tau+\beta}\left[\tau\mathbf{x}^k + \mathbf{z}^k + \beta\mathbf{y}^{k+1} - A^\top(A\mathbf{x}^k - \mathbf{c})\right]\right\}$.
        3) Update $z^k$ via  $\mathbf{z}^{k+1} = \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$.
    End Do
```

Remark 2.1. To ensure convergence, we require the proximal parameter τ to be sufficiently large — viz. $\tau > \rho(B^\top B)$ for Algorithm 1 and $\tau > \rho(A^\top A)$ for Algorithm 2, respectively. For some applications such as image deblurring problems, the estimation of $\rho(A^\top A)$ and $\rho(B^\top B)$ is not difficult. Indeed, when A is a blurring operator we usually normalize (i.e. require $\|A\|_\infty = 1$) to ensure that the blurred image remains within the given dynamic range — e.g. $\rho(A^\top A) = \|A\|^2 \le \|A\|_\infty \|A\|_1 = 1$ for a symmetric blurring A. For a discrete gradient operator B, we have the discrete Laplacian operator $B^\top B$, and hence $\rho(B^\top B) \le 8$ from Gerschgorin's theorem [35]. Even for generic operators, if we assume periodic boundary conditions then the operators can be diagonalized by Fast Fourier Transform (e.g. See [6,26]), so their spectral radius can be computed easily. One can also employ some backtracking technique to identify an appropriate τ — e.g. see [2].

To clarify our main idea, let us fix the value of the penalty parameter β as a positive constant in our proposed algorithms. For strategies that adjust this parameter dynamically, see Ref. [20] for example.

3. Convergence

In this Section, we establish the convergence of our two proposed algorithms. We only prove the convergence of Algorithm 1 and omit the proof for Algorithm 2, since it is quite similar. We assume that a minimizer of (1.1) exists, as it does if $\ker(A^\top A)\cap\ker(B^\top B)=\emptyset$, when $A^\top A + \lambda^2 B^\top B$ is positive definite such that (1.1) is strictly convex and admits a unique solution $\mathbf{x}^*$ for arbitrarily given $\mathbf{c}$ and $\mathbf{d}$ — e.g. see Ref. [4]. For image deblurring problems, A and B respectively represent a blurring operator and a regularization operator — and since a blurring operator is an integral operator (a low-pass filter) but 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 note 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

$$\begin{cases} A^T(Ax^* - c) - z^* = 0, \\ (y' - y^*)^T(\lambda^2 B^\top(By^* - d) + z^*) \geq 0, \ \ \forall\, y' \in \Omega\,, \\ x^* - y^* = 0\,. \end{cases} \tag{3.1}$$

For notational convenience, let us write $\mathbf{w}^\top = (\mathbf{x}^\top, \mathbf{y}^\top, \mathbf{z}^\top)$ and $\mathbf{v}^\top = (\mathbf{y}^\top, \mathbf{z}^\top)$, Υ^* for the solution set of (3.1), and for $\mathbf{w}^* = (\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*) \in \Upsilon^*$ also write $\mathbf{v}^* = (\mathbf{y}^*, \mathbf{z}^*)$. Since (1.1) admits a solution, Υ^* is nonempty — indeed, Υ^* is convex due to the monotonicity of (3.1). Let

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

a matrix which for Algorithm 1 is obviously positive definite when $\tau > \rho(B^\top B)$. Let us now present several lemmas, in proceeding to establish the convergence of Algorithm 1.

Lemma 3.1. *Let*

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

Then $\mathscr{F}$ is monotone — i.e. $(\mathbf{w}' - \mathbf{w})^\top(\mathscr{F}(\mathbf{w}') - \mathscr{F}(\mathbf{w})) \geq 0\,,\ \forall\, \mathbf{w}'$ *and* $\mathbf{w} \in \Upsilon$.

Proof. The proof is elementary, and therefore omitted. □

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

Lemma 3.2. *Let $\{\mathbf{w}^k\}$ be the sequence generated by Algorithm* 1. *Then* $\forall \mathbf{w}' \in \Upsilon$,

$$\begin{aligned} (\mathbf{w}' - \mathbf{w}^{k+1})^\top \Bigg\{ & \mathscr{F}(\mathbf{w}^{k+1}) + \begin{pmatrix} \beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ -\beta(\mathbf{y}^{k+1} - \mathbf{y}^k) \\ \mathbf{0} \end{pmatrix} \\ & - \begin{pmatrix} 0 & 0 & 0 \\ 0 & (\beta + \lambda^2\tau)I - \lambda^2 B^\top B & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix} (\mathbf{w}^k - \mathbf{w}^{k+1}) \Bigg\} \geq 0\,. \end{aligned} \tag{3.3}$$

Proof. From (2.5), (2.10) and $\mathbf{z}^{k+1} = \mathbf{z}^k - \beta(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$, $\forall \mathbf{w}' \in \lambda$ we have the variational inequalities

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

Recalling the definition of $\mathscr{F}$, then $\forall \mathbf{w}' \in \Upsilon$ we have

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

which immediately implies (3.3). □

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 as defined in* (3.2)*. Then*

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

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

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

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

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

which combined with $\mathbf{v}^{k+1}-\mathbf{v}^*=\mathbf{v}^k-\mathbf{v}^*-(\mathbf{v}^k-\mathbf{v}^{k+1})$ yields (3.5). □

From the preceding lemmas we derive some important properties of $\{\mathbf{v}^k\}$, as summarized in the following lemma.

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

(i) $\lim_{k\to\infty}\|\mathbf{v}^k-\mathbf{v}^{k+1}\|_G=0$*;*

(ii) $\{\mathbf{v}^k\}$ *is a bounded sequence*; *and*

(iii) $\|\mathbf{v}^k-\mathbf{v}^*\|_G^2$ *is non-increasing and thus converges.*

Proof. On writing $\mathbf{v}^{k+1}=\mathbf{v}^k-(\mathbf{v}^k-\mathbf{v}^{k+1})$, from (3.5) we have

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

If we denote

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

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 this inequality into (3.6) yields

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

where

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

The statements of the lemma trivially follow from (3.7). □

The convergence of Algorithm 1 is now covered in the following theorem:

Theorem 3.1. *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

$$\lim_{k\to\infty} \|\mathbf{y}^k - \mathbf{y}^{k+1}\| = 0\,, \qquad \lim_{k\to\infty} \|\mathbf{z}^k - \mathbf{z}^{k+1}\| = 0\,. \tag{3.8}$$

From (ii) of Lemma 3.4, $\{\mathbf{v}^k\}$ has at least one accumulation point, which we may denote 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. $\mathbf{y}^{k_j} \to \mathbf{y}^\infty$ and $\mathbf{z}^{k_j} \to \mathbf{z}^\infty$. 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 $\mathbf{y}^{k_j} \to \mathbf{y}^\infty$ and $\mathbf{z}^{k_j} \to \mathbf{z}^\infty$, we have $\mathbf{y}^{k_j} - \mathbf{y}^{k_j - 1} \to \mathbf{0}$ and $\mathbf{z}^{k_j} - \mathbf{z}^{k_j - 1} \to \mathbf{0}$ such that

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

so $(x^\infty, y^\infty, z^\infty)$ 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, on substituting (3.8) into (3.4) we have

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

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

$$\|\mathbf{v}^{k+1} - \mathbf{v}^\infty\| \leq \|\mathbf{v}^k - \mathbf{v}^\infty\|, \;\; \forall\, k \geq 0\,, \tag{3.11}$$

so $\mathbf{v}^\infty$ is the only accumulation point of the sequence $\{\mathbf{v}^k\}$, and hence $\{\mathbf{v}^k\}$ 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). □

4. Numerical Experiments

In this Section, we test our proposed linearized ADMM algorithms on the image deblurring problem

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

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 [24] (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 projection methods and some existing Newton-type methods. All the codes were

Figure 1: Original images

written with MATLAB 7.8 (R2009a), and run on a T6500 notebook with the Intel Core 2 Duo CPU at 2.1 GHz and 2GB of memory.

In our comparison, we test the same 256 × 256 images as in Ref. [24] — viz. the Eagle, Church, Satellite and Bridge images in Fig. 1. Accordingly, $m = n = 65,536$ in model (1.1) for these images. As in Ref. [24], 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. They are therefore diagonalizable by the 2D discrete Fourier transforms [6], and hence our Algorithms 1 and 2 involve $O(n \log n)$ 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 η is the noise level. The bound constraints are set to $l_i = 0$ and $u_i = 255$ for all $i = 1, \cdots, n$. We employ the MATLAB scripts `A = fspecial(’average’,alpha)` and `C = imfilter(X,A,’circular’,’conv’) +` η`*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 the Algorithms 1 and 2 when the relative change $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|/\|\mathbf{x}^{k-1}\| \le 10^{-4}$. We measured the quality of restoration by the peak signal-to-noise ratio (PSNR) in decibles (dB) defined by

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

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 numerical experiments, we set $\beta = 0.1$ for both Algorithms 1 and 2. We recall from Theorem 3.1 that the convergence of Algorithm 1 (Algorithm 2, respectively) is ensured when $\tau > \rho(B^\top B)$ ($\tau > \rho(A^\top A)$, respectively), so 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). 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 Ref. [24]. In Table 1, we report the performance of RN, AS, Algorithm 1 (Alg1) and Algorithm 2 (Alg2) for different levels of

Table 1: Numerical comparison of RN, AS, Algorithm 1 and Algorithm 2.

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

noise η and blur size `alpha`. We also include the computing time in seconds (denoted by "Time(s)"), the 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 was terminated. It is notable that the Newton-type methods RN and AS both require an interior point inside $[0,255]^n$ as the initial iterate. We applied the method in Ref. [24] to generate their initial iterate — i.e. we solved the unconstrained version of (4.1)

$$\min_{\mathbf{x}\in\mathbb{R}^n}\left\{\frac{1}{2}\|A\mathbf{x}-\mathbf{c}\|^2+\frac{\lambda^2}{2}\|B\mathbf{x}\|^2\right\}, \tag{4.2}$$

and then projected the solution onto the box $[1,254]^n$ to do so. Note that solving (4.2) amounts to solving its normal equation $(A^\top A+\lambda^2 B^\top B)\mathbf{x}=A^\top\mathbf{c}$. In contrast, our Algorithms 1 and 2 can use any arbitrary image as the initial iterate, and in our experiments we simply started their iterations from the blurred image $\mathbf{c}$. In the last column of Table 1, the "PJ" column, we also report the PSNR of the projected solution of (4.2). We solved (4.2) by the least squares method, and then projected the solution onto the box $[0,255]^n$. For each test case, we repeated the experiment three times and report the average performance.

The efficiency of our Algorithms 1 and 2 are shown clearly in Table 1, where the best results are shown in boldface. We emphasize that the PSNR obtained by any of the four algorithms for the constrained model (4.1) is 0.2 dB higher at 2.2 dB than that obtained by the projected solution of the unconstrained model (4.2) — cf. the "PJ" column in the table, so it pays to solve the constrained model (4.1). Secondly, we see that Algorithm 1 gives the smallest objective function values of all four methods, and it is the fastest. Algorithm 1 involves matrix-vector multiplication of $B^\top B$ that requires $O(n)$ operations, whereas Algorithm 2 involves matrix-vector multiplication of $A^\top A$ in $\mathrm{O}(n\log n)$ operations, so the iteration cost for Algorithm 1 is notably less.

In regard to the PSNR, Algorithm 1 is either the best or within 0.15 dB from the best, where the difference is due to the fact that we fixed $\lambda = 0.1$ for all cases. The regularization parameter λ should be chosen according to the noise level. For Algorithms 1 and 2 and "PJ", we tested a finite number of choices of λ by trial and error, and respectively chose the best λ value (denoted by λ^*) to achieve the highest PSNR. The results are reported

Table 2: Numerical comparison of Algorithm 1 and Algorithm 2 with best λ^*.

Image	η	λ^*		PSNR (dB)		Time (s)		λ^*	PJ
`alpha`		Alg1	Alg2	Alg1	Alg2	Alg1	Alg2		(dB)
Eagle	3	0.16	0.16	37.65	37.64	0.47	0.92	0.17	33.20
3	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
3	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	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	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

in Table 2, where it appears that (with a well tuned λ) Algorithm 1 can actually attain a slightly better PSNR than Algorithm 2. Again, Algorithm 1 is much faster than Algorithm 2, and 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 two algorithms, we used the MATLAB script `fspecial` to generate four more different blurred images and compared their restoration results. We added the '`average`' blur to the satellite image with `alpha` $=$ 3 and noise level $\eta = 4$, and to the church image with `alpha` $= 5$ and $\eta = 2$. For the eagle image, we adopted the '`motion`' blur kernel with $len = 5$, $theta = 15$ and the noise level $\eta = 2$. For the bridge image, we applied the '`Gaussian`' blur kernel with $hsize = 5$, $\sigma = 4$ and the noise level $\eta = 4$. The restored images are shown in Fig. 2, where both timing and accuracy are given. We again see that Algorithm 1 is faster than the other methods, and its PSNR is either the best or differs from the best by no more than 0.01dB.

5. Concluding Remarks

This paper provides a novel approach to solving constrained linear least-squares problems, by combining linearization techniques with the alternating direction method of multipliers (ADMM). This approach belongs to the category of inexact ADMM, where approximate subproblem solutions are obtained by linearizing some quadratic terms involved. Since the ADMM is a first-order method that generally requires more outer iterations compared to second-order methods, of major concern in making the ADMM efficient is to alleviate the difficulty of obtaining solutions in the inner iterations. The proposed linearization technique does this well, because the linearized inner subproblems have closed-form solutions.

We have proven the convergence of the proposed linearized ADMM approach, and applied it to solve some image deblurring problems. Compared to Newton-type methods, the linearized ADMM approach is more efficient in both speed and restored image quality. Moreover, the linearized ADMM approach does not require a good initial guess to start the iteration, in contrast to Newton-type methods that are usually sensitive to the choice of the

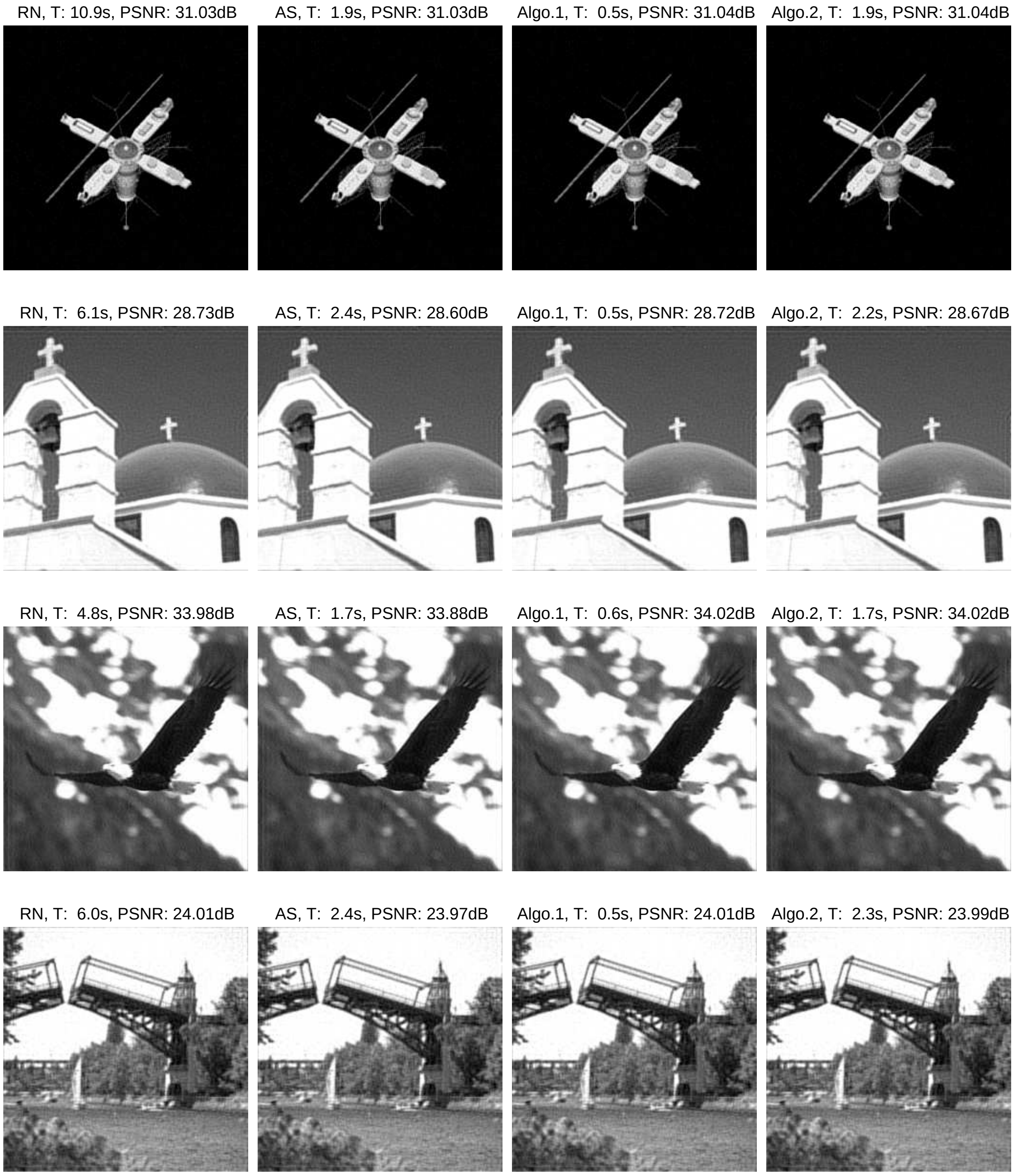
RN, T: 10.9s, PSNR: 31.03dB
AS, T: 1.9s, PSNR: 31.03dB
Algo.1, T: 0.5s, PSNR: 31.04dB
Algo.2, T: 1.9s, PSNR: 31.04dB
RN, T: 6.1s, PSNR: 28.73dB
AS, T: 2.4s, PSNR: 28.60dB
Algo.1, T: 0.5s, PSNR: 28.72dB
Algo.2, T: 2.2s, PSNR: 28.67dB
RN, T: 4.8s, PSNR: 33.98dB
AS, T: 1.7s, PSNR: 33.88dB
Algo.1, T: 0.6s, PSNR: 34.02dB
Algo.2, T: 1.7s, PSNR: 34.02dB
RN, T: 6.0s, PSNR: 24.01dB
AS, T: 2.4s, PSNR: 23.97dB
Algo.1, T: 0.5s, PSNR: 24.01dB
Algo.2, T: 2.3s, PSNR: 23.99dB

Figure 2: From left to right: Results recovered by RN, AS, Algorithms 1 and 2, respectively."T" represents running time.

initial iterate.

This research may be extended to develop inexact ADMMs for more complex problems in image processing, such as the multiframe deblurring problem where the objective function takes the form

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

Acknowledgments

Our research is supported in part by HKRGC Grant No. CUHK400510 and the CUHK Direct Allocation Grant 2060408, the Scientific Research Foundation of the Nanjing University of Posts and Telecommunications (NY210049), and the Hong Kong General Research Fund: HKBU 203311.

References

[1] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8, pp. 141–148, 1988.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-Thresholding algorithms for inner inverse problems*, SIAM J. Imag. Sci., 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*, Numer Linear Algebra Appl., 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 Probl., 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 (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*, SIAM J. Imag. Sci., 4, pp. 807–826, 2011.

[9] T. Coleman and Y. Li, *An interior trust-region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6, pp. 418–445, 1996.

[10] M. Ehrgott and 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, pp. 17–40, 1976.

[14] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems, M. Fortin and R. Glowinski, North Holland, Amsterdam, Holland, pp. 299–331, 1983.

[15] R. Glowinski and A. Marrocco, *Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de Dirichlet nonlineaires*, Rev. Francaise dAut. Inf. Rech. Oper., 2, pp. 41–76, 1975.

[16] R. Glowinski and P. Le Tallec, Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989.

[17] T. Goldstein and S. Osher, *The split Bregman method for L_1-Regularized Prolbems*, SIAM J. Imag. Sci., 2, pp. 323–343, 2009.

[18] R. Gonzalez and R. Woods, Digital Image Processing, 3rd ed., Prentice Hall, NJ, 2008.

[19] 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.

[20] B. S. He, H. Yang and S. L. Wang, *Alternating direction method with self-adaptive penalty parameters for monotone variational inequality*, J. Optim. Theory Appl., 106, pp. 337–356, 2000.

[21] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4, pp. 303–320, 1969.

[22] 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.

[23] C. L. Lawson and R. J. Hanson, Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[24] B. Morini, M. Porcelli and R. H. Chan, *A reduced Newton method for constrained linear least-squares problems*, J. Comput. Appl. Math., 233, pp. 2200–2212, 2010.

[25] S. Morigi, L. Reichel, F. Sgallari and F. Zama, *An iterative method for linear discrete ill-posed problems with box constrints*, J. Comput. Appl. Math., 198, pp. 505–520, 2007.

[26] 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.

[27] M. K. Ng, F. Wang and X. M. Yuan, *Fast minimization methods for solving constrained total-variation superresolution image reconstruction*, Multi. Syst. Signal Proces., 22, pp. 259-286, 2011.

[28] 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.

[29] 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.

[30] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher (Ed.), Academic Press, New York, pp. 283–298, 1969.

[31] M. Rojas and T. Steihaug, *An interior-point trust-region-based method for large-scale nonnegative regularization*, Inverse Probl., 18, pp. 1291–1307, 2002.

[32] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60, pp. 259–268, 1992.

[33] S. Setzer, *Split Bregman algorithm, Douglas-Rachford splitting, and frame shrinkage*, Lect. Notes Comput. Sci., 5567, pp. 464–476, 2009.

[34] S. Setzer, G. Steidl and T. Tebuber, *Deblurring Poissonian images by split Bregman techniques*, J. Vis. Commun. Image Repres., 21, pp. 193–199, 2010.

[35] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Texts in Applied Mathematics, No. 12, Springer, 2007.

[36] M. Tao and X. M. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM J. Optim., 21(1), pp. 57–81, 2011.

[37] A. Tikhonov and V. Arsenin, Solution of Ill-Posed Problems, Winston, Washington, DC, 1977.

[38] P. Tseng, *Applications of splitting algorithm to decomposition in convex programming and vartiational inequalities*, SIAM J. Control Optim., 29, pp. 119–138, 1991.