MATHEMATICS OF COMPUTATION
Volume 82, Number 281, January 2013, Pages 301–329
S 0025-5718(2012)02598-1
Article electronically published on March 28, 2012

LINEARIZED AUGMENTED LAGRANGIAN AND ALTERNATING DIRECTION METHODS FOR NUCLEAR NORM MINIMIZATION

JUNFENG YANG AND XIAOMING YUAN

Abstract. The nuclear norm is widely used to induce low-rank solutions for many optimization problems with matrix variables. Recently, it has been shown that the augmented Lagrangian method (ALM) and the alternating direction method (ADM) are very efficient for many convex programming problems arising from various applications, provided that the resulting subproblems are sufficiently simple to have closed-form solutions.

In this paper, we are interested in the application of the ALM and the ADM for some nuclear norm involved minimization problems. When the resulting subproblems do not have closed-form solutions, we propose to linearize these subproblems such that closed-form solutions of these linearized subproblems can be easily derived.

Global convergence results of these linearized ALM and ADM are established under standard assumptions. Finally, we verify the effectiveness and efficiency of these new methods by some numerical experiments.

1. Introduction

Let $X^* \in \mathcal{R}^{m\times n}$ be an unknown matrix and $b = \mathcal{A}X^* \in \mathcal{R}^p$ $(p < mn)$, where $\mathcal{A}$ stands for a linear mapping from $\mathcal{R}^{m\times n}$ to $\mathcal{R}^p$. We are often asked to reconstruct X^* from b with $\mathcal{A}$ given. Clearly, without further conditions this task is not trackable since $p < mn$. However, when X^* is a low-rank matrix, this reconstruction becomes possible via solving the following convex programming model:

$$\min_{X\in\mathcal{R}^{m\times n}} \{\|X\|_* : \mathcal{A}X = b\}, \tag{1.1}$$

where $\|\cdot\|_*$ is the so-called nuclear norm (also known as trace norm or Ky Fan norm) defined as the sum of all singular values. Note that the nuclear norm is the convex envelope of the rank function over the unit ball under the spectral norm (see, e.g., [14]), and it is a widely used surrogate of the rank function to induce low-rank solutions in various areas such as machine learning, statistics, engineering, etc. When $\mathcal{A}$ is a sampling operator collecting a fraction of entries of a matrix,

Received by the editor July 30, 2010 and, in revised form, April 18, 2011 and August 9, 2011.
2010 *Mathematics Subject Classification.* Primary 90C25, 90C06, 65K05.
Key words and phrases. Convex programming, nuclear norm, low-rank, augmented Lagrangian method, alternating direction method, linearized.
The work of the first author was supported by the Natural Science Foundation of China NSFC-11001123 and the Fundamental Research Funds for the Central Universities (Grant No. 1117020305).
The work of the second author was supported by the Hong Kong General Research Fund HKBU-202610.

(1.1) reduces to the well-known matrix completion problem. We refer to [8, 9] for some breakthrough results on the matrix completion problem, and [40] for recent development on (1.1) with a generic linear operator $\mathcal{A}$.

In practice, b is often obtained through hardware implementation, and it usually suffers from a nontrivial level of noise. That is, $b = \mathcal{A}X^* + \omega$, where ω contains measurement errors dominated by certain normal distribution. With the consideration of noise, (1.1) is often relaxed to the nuclear norm regularized least squares problem

$$\min_{X\in\mathcal{R}^{m\times n}} \|X\|_* + \frac{1}{2\mu}\|\mathcal{A}X - b\|_2^2, \tag{1.2}$$

or its constrained alternative

$$\min_{X\in\mathcal{R}^{m\times n}} \left\{\|X\|_* : \|\mathcal{A}X - b\|_2 \le \delta\right\}, \tag{1.3}$$

where $\mu > 0$ and $\delta > 0$ are parameters reflecting the level of noise. Based on their respective optimality conditions, the models (1.2) and (1.3) are theoretically related in the sense that solving one of them can determine a value of the noise parameter for the other such that these two models share a common solution. In the area of machine learning, (1.2) is of particular interest. For example, it is shown in [1, 2, 36, 38] that certain multi-task learning problems with least squares loss functions can be exactly captured by (1.2). On the other hand, the model (1.3) is often preferred when a reasonable estimation of the noise level is available; see [31].

In practice, it is not trivial to solve (1.1)-(1.3), and there appears to be a growing interest in developing customized algorithms, particularly for large-scale cases of these nuclear norm involved problems. We now briefly review some influential approaches to these problems. First, the convex problems (1.1)-(1.3) can be easily reformulated into semidefinite programming (SDP) problems (see, e.g., [14, 44]), and thus generic SDP solvers based on interior-point methods such as SeDuMi [45] and SDPT3 [51] are in principle applicable. However, as pointed out in [26, 38, 40], the interior-point approach is prohibitively inefficient for large-scale (or even medium) cases of these problems. In [40], a projected subgradient approach is suggested to solve (1.1), whose computation at each iteration is dominated by one singular value decomposition (SVD). The projected subgradient approach is easily implementable, and it can be applied to large-scale cases of (1.1). However, this projected subgradient method suffers from slow convergence, especially when high accuracy is required. In [40], the authors also develop the UV-parametrization approach to general low-rank matrix reconstruction problems. Specifically, the low-rank matrix X is decomposed into the form $UV^\top$, where $U \in \mathcal{R}^{m\times r}$ and $V \in \mathcal{R}^{n\times r}$ are tall and thin matrices. We refer to [54] for a similar approach. This UV-parametrization approach is capable of avoiding the computation of SVD (see, e.g., [54]), and it benefits from the reduction of dimensionality from mn to $(m + n)r$ (normally, $r \ll \min(m, n)$ for optimization problems with the low-rank concern). However, the parameter r is not known a priori for most of applications, and it has to be estimated or dynamically adjusted, which might be difficult to realize. The classical augmented Lagrangian method (also known as the method of multipliers, see, e.g., [25, 39, 46]) is also discussed in [40] for low-rank matrix recovery problems based on this UV-parametrization.

In the context of the matrix completion problem where the linear operator $\mathcal{A}$ is a sampling (or projection/restriction) operator (see (1.4) for details), a singular

value thresholding approach is proposed in [4] for solving a regularized version of (1.1), and a fixed point continuation scheme is proposed in [32] for solving (1.2). Moreover, some accelerated proximal gradient algorithms based on Nesterov's work [33, 34] are developed in [26, 49] for solving (1.2). In particular, the method in [49] terminates in $O(1/\sqrt{\varepsilon})$ iterations to attain an ε-optimal solution. The method in [26] achieves the convergence rate $O(1/k^2)$ for a more general case of (1.2) where the least squares loss function is replaced by a generic smooth loss function. In [31], a proximal point algorithmic framework is proposed for solving a generalized constrained nuclear norm minimization problem. In [38], the authors first show that a special case of (1.2) arising from multi-task learning (see (6.1) for details) is reducible to the case where the coefficient matrix has full column rank. Then, they propose several gradient type methods to solve both (6.1) and its dual, with the effort of reducing the computation of SVD.

Recently, it has been shown in the literature that the augmented Lagrangian method (ALM) [25, 39] and the alternating direction method (ADM) [17] are very efficient for some convex programming problems arising from various applications, provided that the resulting subproblems are simple enough to have closed-form solutions or can be easily solved up to high precisions. Here, we mention a few of such applications: image processing [13, 19, 35, 43], compressive sensing [55], SDP [47, 53] and statistics [22, 29, 48]. In particular, the ADM is applied to solve nuclear norm based matrix completion problems in [10] where the sampling operator $\mathcal{A}$ is in the form of

$$\mathcal{A}X = X_\Omega. \tag{1.4}$$

Here, $\Omega \subset \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ is an index set reflecting known entries of X^*, and X_Ω is a vector formed by the components of X with indices in Ω. Due to the simplicity of the linear operator $\mathcal{A}$, all the ADM subproblems of the matrix completion problem can be solved exactly by explicit formulas; see [10] for details. In general, the ADM derived in [10] (and those in Section 2 of the present paper) can be viewed as the split Bregman method discussed in [7, 19] for the general problems with ℓ_1-like regularization where the augmented Lagrangian functions are minimized by only one round of alternating minimization.

In this paper, we first focus on the special case of (1.1)-(1.3) where $\mathcal{A}\mathcal{A}^* = \mathcal{I}$. Here and hereafter $\mathcal{A}^*$ and $\mathcal{I}$ represent the adjoint of $\mathcal{A}$ and the identity operator, respectively. In particular, we show that when the ADM is applied to this case, the resulting subproblems for (1.1)-(1.3) all have closed-form solutions. We then concentrate on the general case where $\mathcal{A}\mathcal{A}^* \neq \mathcal{I}$. The ALM and the ADM are also applicable to (1.1)-(1.3), after some easy reformulations. However, when the ALM and the ADM are applied to (1.1)-(1.3) with a generic linear operator $\mathcal{A}$, some of the resulting subproblems no longer have closed-form solutions, and the efficiency of the ALM and the ADM depends heavily on how to solve these harder subproblems. We hence propose to linearize those harder subproblems such that closed-form solutions of the linearized subproblems can be achieved. Consequently, linearized ALM and ADM are developed for solving (1.1)-(1.3) with a generic linear operator $\mathcal{A}$. The efficiency of these linearized methods is well illustrated by some numerical experiments including comparisons with some existing efficient methods.

Throughout this paper, we use the following notation. We let $\langle \cdot, \cdot \rangle$ be the standard inner product in a finite dimensional Euclidean space, $\|\cdot\|$ be the 2-norm, and $\|\cdot\|_F$ be the Frobenius norm for matrix variables. The transpose of a real matrix

is denoted by the superscript "$\top$". The projection operator under the Euclidean distance measure is denoted by $\mathcal{P}$. Other notation will be introduced as it occurs.

The rest of this paper is organized as follows. In Section 2, we apply the ADM to solve (1.1)-(1.3) with $\mathcal{A}\mathcal{A}^* = \mathcal{I}$. Sections 3 and 4 concentrate on (1.1)-(1.3) with a generic linear operator $\mathcal{A}$. Specifically, in Section 3 we present linearized ALM for (1.1) and establish its global convergence. Then, in Section 4, we extend the same linearization idea to solve (1.2) and (1.3), and then derive linearized ADMs. Global convergence of the linearized ADMs are also established. In Section 5, we clarify the connections of the linearized ALM and ADMs with some existing work in the literature. Numerical results, including comparisons with some existing methods, are reported in Section 6. Finally, conclusions are drawn in Section 7.

2. ADMs for (1.1)-(1.3)

In this section, we consider the special case of (1.1)-(1.3) with $\mathcal{A}$ satisfying $\mathcal{A}\mathcal{A}^* = \mathcal{I}$, which has wide applications such as the aforementioned matrix completion problem. We show that when the ADM [17] is applied to this special case, all the resulting subproblems have closed-form solutions. We start this section with some preliminaries which are convenient for the presentation of algorithms later.

For $\delta \geq 0$, we define

$$\mathbf{B}_\delta := \{U \in \mathcal{R}^{m\times n} : \|\mathcal{A}U - b\| \leq \delta\}. \tag{2.1}$$

In particular, $\mathbf{B}_0 = \{U \in \mathcal{R}^{m\times n} : \mathcal{A}U = b\}$.

For any $\alpha > 0$, it is easy to verify that

$$(\mathcal{I} + \alpha\mathcal{A}^*\mathcal{A})^{-1} = \mathcal{I} - \frac{\alpha}{1+\alpha}\mathcal{A}^*\mathcal{A}, \tag{2.2}$$

where $(\mathcal{I} + \alpha\mathcal{A}^*\mathcal{A})^{-1}$ denotes the inverse operator of $\mathcal{I} + \alpha\mathcal{A}^*\mathcal{A}$.

For $\delta > 0$ and $Y \in \mathcal{R}^{m\times n}$, the projection of Y onto $\mathbf{B}_\delta$ is given by

$$\mathcal{P}_{\mathbf{B}_\delta}(Y) = Y + \frac{\eta}{\eta+1}\mathcal{A}^*\left(b - \mathcal{A}Y\right), \tag{2.3}$$

where

$$\eta = \max\{\|\mathcal{A}Y - b\|/\delta - 1, 0\}. \tag{2.4}$$

In particular,

$$\mathcal{P}_{\mathbf{B}_0}(Y) = Y + \mathcal{A}^*(b - \mathcal{A}Y). \tag{2.5}$$

To see (2.3), we have that

$$\mathcal{P}_{\mathbf{B}_\delta}(Y) = \mathrm{argmin}_{X\in\mathcal{R}^{m\times n}}\{\|X - Y\|_F^2 : \|\mathcal{A}X - b\| \leq \delta\},$$

whose solution is characterized by the following system (deriving the KKT condition of the above minimization problem):

$$\begin{aligned} X - Y + \eta\mathcal{A}^*(\mathcal{A}X - b) &= 0,\\ \|\mathcal{A}X - b\| \leq \delta,\ \eta &\geq 0,\\ \eta\left(\|\mathcal{A}X - b\| - \delta\right) &= 0. \end{aligned}$$

Obviously, $X := \mathcal{P}_{\mathbf{B}_\delta}(Y)$ and η defined respectively in (2.4) and (2.3) satisfy the above system.

Similarly, consider the problem:

$$\min_{X\in\mathcal{R}^{m\times n}}\{\|X - Y\|_F^2 : \mathcal{A}X = b\},$$

whose KKT system is given by

$$X - Y - \mathcal{A}^* h = 0 \text{ and } \mathcal{A}X = b.$$

Then, it is easy to check that $X = \mathcal{P}_{\mathbf{B}_0}(Y)$ defined in (2.5) and $h := b - \mathcal{A}Y$ satisfy the above system. Hence, (2.5) is justified.

With given $Y \in \mathcal{R}^{m\times n}$ and $\delta > 0$, let $Y = U\Sigma V^\top$ be the SVD of Y and I be the identity matrix. We define the following "shrinkage" operator

$$\mathcal{D}_\delta(Y) := U(\Sigma - \delta I)_+ V^\top, \tag{2.6}$$

where $(a)_+ := \max\{a, 0\}$. Then, it can be shown that (see, e.g., [4, 32])

$$\mathcal{D}_\delta(Y) = \operatorname{argmin}_{X\in\mathcal{R}^{m\times n}} \big\{\delta\|X\|_* + \frac{1}{2}\|X - Y\|_F^2\big\}. \tag{2.7}$$

Next, we start to derive the ADMs for (1.1)-(1.3), in the order of (1.1), (1.3) and (1.2).

By introducing an auxiliary variable $Y \in \mathcal{R}^{m\times n}$, (1.1) is equivalently transformed to

$$\min_{X,Y}\{\|X\|_* : X = Y,\ Y \in \mathbf{B}_0\}. \tag{2.8}$$

The augmented Lagrangian function of (2.8) is given by

$$\mathcal{L}(X, Y, Z, \beta) := \|X\|_* - \langle Z, X - Y\rangle + \frac{\beta}{2}\|X - Y\|_F^2, \tag{2.9}$$

where Z is the Lagrange multiplier and $\beta > 0$ is the penalty parameter for the violation of the linear constraints. For simplicity, throughout this paper we assume that $\beta > 0$ is fixed.

Overall speaking, the ADM [17] is a practical variant of the ALM [25, 39] for linearly (and possibly other simple set) constrained convex programming problems with separable objective functions. For the extensive study of ADM in the context of convex programming and variational inequalities, we refer to, e.g., [6, 12, 15, 16, 18, 21, 27, 50]. When it is applied to solve (2.8), the ADM minimizes $\mathcal{L}(X, Y, Z, \beta)$ with respect to X and Y in an alternating order at each iteration, differing from the ALM which minimizes $\mathcal{L}(X, Y, Z, \beta)$ with respect to X and Y simultaneously. More specifically, given Y^k, Z^k and β, the iterative scheme of ADM for (2.8) reads

$$X^{k+1} = \arg\min_X \mathcal{L}(X, Y^k, Z^k, \beta), \tag{2.10a}$$

$$Y^{k+1} = \arg\min_{Y\in\mathbf{B}_0} \mathcal{L}(X^{k+1}, Y, Z^k, \beta), \tag{2.10b}$$

$$Z^{k+1} = Z^k - \beta(X^{k+1} - Y^{k+1}). \tag{2.10c}$$

It is easy to see that the X-subproblem (2.10a) is reducible to a problem in the form of (2.7), and it thus can be solved by the shrinkage operator (2.6). On the other hand, the Y-subproblem (2.10b) amounts to a projection problem onto $\mathbf{B}_0$, and it thus can be solved by (2.5). Specifically, simple computation shows that the new iterate $(X^{k+1}, Y^{k+1}, Z^{k+1})$ in (2.10) can be explicitly represented by

$$X^{k+1} = \mathcal{D}_{1/\beta}(Y^k + Z^k/\beta), \tag{2.11a}$$

$$Y^{k+1} = \mathcal{P}_{\mathbf{B}_0}(X^{k+1} - Z^k/\beta), \tag{2.11b}$$

$$Z^{k+1} = Z^k - \beta(X^{k+1} - Y^{k+1}), \tag{2.11c}$$

which shows that all the resulting subproblems are simple enough to have closed-form solutions when the ADM is applied to (1.1).

The application of ADM to (1.3) is completely analogous to that of (1.1). By introducing Y, (1.3) is equivalent to (2.8), but $\mathbf{B}_\delta$ rather than $\mathbf{B}_0$. The corresponding augmented Lagrangian function is the same as that defined in (2.9), and the ADM scheme for (1.3) is identical to (2.10) except that $\mathbf{B}_0$ is replaced by $\mathbf{B}_\delta$. Eventually, the ADM iterative formulas for (1.3) is the same as (2.11) except that $\mathcal{P}_{\mathbf{B}_0}$ in (2.11b) is replaced by $\mathcal{P}_{\mathbf{B}_\delta}$. Since $\mathcal{P}_{\mathbf{B}_\delta}$, for $\delta > 0$, can be computed easily by (2.3)-(2.4), all the resulting subproblems again have closed-form solutions when the ADM is applied to (1.3).

The treatment of (1.2) is also easy. Clearly, (1.2) is equivalent to

$$\min_{X,Y} \left\{ \|X\|_* + \frac{1}{2\mu}\|\mathcal{A}Y - b\|^2 : X = Y \right\}, \tag{2.12}$$

which, as a result of introducing Y, has a separable objective function. The augmented Lagrangian function of (2.12) is

$$\mathcal{L}_U(X, Y, Z, \beta) := \|X\|_* + \frac{1}{2\mu}\|\mathcal{A}Y - b\|^2 - \langle Z, X - Y\rangle + \frac{\beta}{2}\|X - Y\|_F^2. \tag{2.13}$$

The algorithmic framework of the ADM for (2.12) has exactly the same form as in (2.10) except that $\mathcal{L}$ is replaced by $\mathcal{L}_U$ and the Y-subproblem is unconstrained. It is easy to see that, for fixed $X = X^{k+1}$, $Z = Z^k$ and β, the minimization of (2.13) with respect to Y is a least squares problem whose normal equation is equivalent to

$$\left(\mathcal{I} + \mathcal{A}^*\mathcal{A}/\beta\mu\right) Y = X^{k+1} + \left(\mathcal{A}^* b/\mu - Z^k\right)/\beta,$$

the solution of which, by using (2.2), is given by

$$Y^{k+1} = \left(\mathcal{I} - \mathcal{A}^*\mathcal{A}/\beta\mu\right)\left(X^{k+1} - Z^k/\beta\right) + (1 - 1/\beta\mu)\mathcal{A}^* b. \tag{2.14}$$

In summary, we have derived ADMs for (1.1)-(1.3) under the condition $\mathcal{A}\mathcal{A}^* = \mathcal{I}$, and all the resulting subproblems are simple enough to have closed-form solutions. It is easy to see that, besides two multiplications of the form $\mathcal{A}X$ and $\mathcal{A}^* y$, the main computation of the derived ADMs at each iteration is one SVD. Since convergence of these ADMs for any fixed $\beta > 0$ is well studied in the literature (see, e.g., [17, 18]), we omit the details here.

On the other hand, when $\mathcal{A}\mathcal{A}^* \neq \mathcal{I}$, some of the resulting ADM subproblems for (1.1)-(1.3) do not have closed-form solutions, and this difficulty could result in inefficiency of the ADM greatly. As we will show, the blend of the linearization and proximal techniques can alleviate this difficulty substantially, and this is the main content of Sections 3 and 4.

3. Linearized ALM for (1.1)

In this section, we present a linearized ALM (LALM for short) for solving (1.1) and analyze its convergence.

3.1. LALM for (1.1).

The augmented Lagrangian function of (1.1) is given by

$$\mathcal{L}(X, \lambda, \beta) := \|X\|_* - \langle \lambda, \mathcal{A}X - b\rangle + \frac{\beta}{2}\|\mathcal{A}X - b\|^2, \tag{3.1}$$

where $\lambda \in \mathcal{R}^p$ is the Lagrange multiplier and $\beta > 0$ is the penalty parameter. Given $\lambda^k \in \mathcal{R}^p$, by applying the classical ALM (see, e.g., [25, 39]) to (1.1), we obtain the following iterative scheme:

$$X^{k+1} = \arg\min_X \mathcal{L}(X, \lambda^k, \beta), \tag{3.2a}$$

$$\lambda^{k+1} = \lambda^k + \beta\left(b - \mathcal{A}X^{k+1}\right). \tag{3.2b}$$

Let $b^k := b + \lambda^k/\beta$. The iterative scheme (3.2) can be rewritten as

$$X^{k+1} = \arg\min_X \|X\|_* + \frac{\beta}{2}\|\mathcal{A}X - b^k\|^2, \tag{3.3a}$$

$$b^{k+1} = b^k + b - \mathcal{A}X^{k+1}. \tag{3.3b}$$

Roughly speaking, it is not necessary to solve the subproblem (3.3a) up to a very high precision in order to ensure the convergence of the iterative scheme (3.3). In fact, to make the ALM (3.3) truly implementable, we pursue the ease of solving this subproblem at each iteration as long as the overall convergence of ALM can be guaranteed. Motivated by this philosophy, we propose to approximate the subproblem (3.3a) by linearizing the quadratic term of its objective function. With this linearization, the resulting approximation to (3.3a) is then simple enough to have closed-form solution. More specifically, we have

$$\frac{1}{2}\|\mathcal{A}X - b^k\|^2 \approx \frac{1}{2}\|\mathcal{A}X^k - b^k\|^2 + \langle g^k, X - X^k\rangle + \frac{1}{2\tau}\|X - X^k\|_F^2, \tag{3.4}$$

where $\tau > 0$ is a proximal parameter, and

$$g^k := \mathcal{A}^*(\mathcal{A}X^k - b^k) = \mathcal{A}^*(\mathcal{A}X^k - b - \lambda^k/\beta) \tag{3.5}$$

is the gradient of $\frac{1}{2}\|\mathcal{A}X - b^k\|^2$ at X^k. Plugging (3.4) into (3.3a) and with simple manipulations, we obtain the following approximation to (3.3a):

$$\min_X \|X\|_* + \frac{\beta}{2\tau}\|X - (X^k - \tau g^k)\|_F^2. \tag{3.6}$$

Obviously, the closed-form solution of (3.6) is obtainable based on (2.7).

In summary, given $\lambda^k \in \mathcal{R}^p$, the proposed LALM for (1.1) generates (X^{k+1}, λ^{k+1}) by the following iterative framework:

$$X^{k+1} = \mathcal{D}_{\tau/\beta}\left(X^k - \tau g^k\right), \tag{3.7a}$$

$$\lambda^{k+1} = \lambda^k + \beta\left(b - \mathcal{A}X^{k+1}\right), \tag{3.7b}$$

where g^k is given in (3.5), and (3.7a) is the solution of (3.6).

3.2. **Convergence analysis.** In this subsection, we establish the global convergence of the LALM scheme (3.7).

Let X^* be an arbitrary solution of (1.1). From standard theory of convex programming, there exists $\lambda^* \in \mathcal{R}^p$ such that the following conditions are satisfied:

$$\mathcal{A}^*\lambda^* \in \partial\|X^*\|_* \text{ and } \mathcal{A}X^* = b, \tag{3.8}$$

where $\partial\|\cdot\|_*$ stands for the subdifferential of the nonsmooth convex function $\|\cdot\|_*$. Specifically, let $X = U\Sigma V^\top$ be the SVD of X, then $\partial\|X\|_*$ is given by (see, e.g., [3, 52])

$$\partial\|X\|_* = \{UV^\top + W : U^\top W = 0,\, WV = 0,\, \|W\|_2 \le 1\},$$

where $\|W\|_2$ represents the operator norm of W, i.e., its largest singular value.

We first prove two lemmas before establishing the convergence for the LALM (3.7).

Lemma 3.1. *Let X^* be an arbitrary solution of (1.1) and $\lambda^* \in \mathcal{R}^p$ be such that the conditions in (3.8) are satisfied. For any fixed $\beta > 0$ and $\tau > 0$, and an arbitrary initial point $\lambda^0 \in \mathcal{R}^p$, the sequence $\{(X^k, \lambda^k)\}$ generated by the LALM scheme (3.7) satisfies*

$$\begin{aligned} \frac{1}{\beta}(\lambda^{k+1} - \lambda^*)^\top(\lambda^k - \lambda^{k+1}) \quad &+ \quad \frac{\beta}{\tau}\langle X^{k+1} - X^*, X^k - X^{k+1}\rangle \\ &\geq \quad (\lambda^k - \lambda^{k+1})^\top \mathcal{A}(X^k - X^{k+1}). \end{aligned} \tag{3.9}$$

Proof. The optimality condition of (3.6) implies that

$$0 \in \partial\|X^{k+1}\|_* + \frac{\beta}{\tau}(X^{k+1} - X^k + \tau g^k). \tag{3.10}$$

It follows from (3.5) and (3.7b) that

$$g^k = \mathcal{A}^*(\mathcal{A}X^k - b - \lambda^k/\beta) = -\frac{1}{\beta}\mathcal{A}^*\lambda^{k+1} - \mathcal{A}^*\mathcal{A}(X^{k+1} - X^k). \tag{3.11}$$

Plugging (3.11) into (3.10), we can rewrite (3.10) as

$$\frac{\beta}{\tau}\left(I - \tau\mathcal{A}^*\mathcal{A}\right)\left(X^k - X^{k+1}\right) + \mathcal{A}^*\lambda^{k+1} \in \partial\|X^{k+1}\|_*. \tag{3.12}$$

Further, considering $\mathcal{A}^*\lambda^* \in \partial\|X^*\|_*$ and the convexity of $\|\cdot\|_*$, there holds

$$\left\langle X^{k+1} - X^*, \frac{\beta}{\tau}\left(I - \tau\mathcal{A}^*\mathcal{A}\right)\left(X^k - X^{k+1}\right) + \mathcal{A}^*(\lambda^{k+1} - \lambda^*)\right\rangle \geq 0,$$

which is easily shown to be equivalent to

$$\begin{aligned} &\left\langle \mathcal{A}(X^* - X^{k+1}), \beta\mathcal{A}(X^k - X^{k+1}) - (\lambda^{k+1} - \lambda^*)\right\rangle \\ &\quad + \frac{\beta}{\tau}\left\langle X^{k+1} - X^*, X^k - X^{k+1}\right\rangle \geq 0. \end{aligned} \tag{3.13}$$

By noting $\mathcal{A}X^* = b$ and the fact that

$$\mathcal{A}(X^* - X^{k+1}) = b - \mathcal{A}X^{k+1} = (\lambda^{k+1} - \lambda^k)/\beta,$$

we can show that (3.9) follows immediately from (3.13). □

Let $u = (X, \lambda) \in \mathcal{R}^{m\times n} \times \mathcal{R}^p$. We define an inner product in $\mathcal{R}^{m\times n} \times \mathcal{R}^p$ by

$$\langle u, v\rangle_G = \frac{\beta}{\tau}\langle X, Y\rangle + \frac{1}{\beta}\lambda^\top\zeta, \tag{3.14}$$

for $v = (Y, \zeta) \in \mathcal{R}^{m\times n} \times \mathcal{R}^p$ and the induced norm by $\|u\|_G^2 = \langle u, u\rangle_G$. We have the following lemma.

Lemma 3.2. *Let (X^*, λ^*) satisfy the conditions in (3.8). For any fixed $\beta > 0$ and an arbitrary initial point $\lambda^0 \in \mathcal{R}^p$, let $\{(X^k, \lambda^k)\}$ be the sequence generated by the LALM scheme (3.7). If $0 < \tau < 1/\rho(\mathcal{A}^*\mathcal{A})$, where $\rho(\mathcal{A}^*\mathcal{A})$ denotes the spectral radius of $\mathcal{A}^*\mathcal{A}$, then we have*

(a) $\|u^k - u^{k+1}\|_G \to 0$;
(b) $\{u^k\}$ *lies in a compact region;*
(c) $\|u^k - u^*\|_G^2$ *is monotonically non-increasing and thus converges.*

Proof. Given the notation defined in (3.14), (3.9) can be rewritten as

$$\langle u^{k+1}-u^*, u^k-u^{k+1}\rangle_G \geq (\lambda^k-\lambda^{k+1})^\top \mathcal{A}(X^k-X^{k+1}).$$

Since $u^{k+1}-u^* = (u^{k+1}-u^k)+(u^k-u^*)$, it follows that

$$\langle u^k-u^*, u^k-u^{k+1}\rangle_G \geq \|u^k-u^{k+1}\|_G^2 + (\lambda^k-\lambda^{k+1})^\top \mathcal{A}(X^k-X^{k+1}). \tag{3.15}$$

From $u^{k+1} = u^k-(u^k-u^{k+1})$, (3.15) and (3.14), it holds that

$$\begin{aligned} &\|u^k-u^*\|_G^2 - \|u^{k+1}-u^*\|_G^2 \\ =\ & 2\langle u^k-u^*, u^k-u^{k+1}\rangle_G - \|u^k-u^{k+1}\|_G^2 \\ \geq\ & 2\|u^k-u^{k+1}\|_G^2 + 2(\lambda^k-\lambda^{k+1})^\top\mathcal{A}(X^k-X^{k+1}) - \|u^k-u^{k+1}\|_G^2 \\ =\ & \frac{\beta}{\tau}\|X^k-X^{k+1}\|_F^2 + \frac{1}{\beta}\|\lambda^k-\lambda^{k+1}\|^2 + 2(\lambda^k-\lambda^{k+1})^*\mathcal{A}(X^k-X^{k+1}). \end{aligned} \tag{3.16}$$

Since $0<\tau<1/\rho(\mathcal{A}^*\mathcal{A})$, it holds that $\delta := 1-\tau\rho(\mathcal{A}^*\mathcal{A})>0$. Let

$$\eta := \frac{1}{\beta(1+\delta)} > 0.$$

Then, from the Cauchy-Schwarz inequality $2a^\top b \geq -\eta\|a\|^2 - \|b\|^2/\eta$, the definitions of η and δ, (3.16) implies that

$$\begin{aligned} &\|u^k-u^*\|_G^2 - \|u^{k+1}-u^*\|_G^2 \\ \geq\ & \frac{\beta}{\tau}\|X^k-X^{k+1}\|_F^2 + \left(\frac{1}{\beta}-\eta\right)\|\lambda^k-\lambda^{k+1}\|^2 - \frac{1}{\eta}\|\mathcal{A}(X^k-X^{k+1})\|^2 \\ \geq\ & \left(\frac{\beta}{\tau}-\frac{\rho(\mathcal{A}^*\mathcal{A})}{\eta}\right)\|X^k-X^{k+1}\|_F^2 + \left(\frac{1}{\beta}-\eta\right)\|\lambda^k-\lambda^{k+1}\|^2 \\ =\ & \frac{\beta\delta^2}{\tau}\|X^k-X^{k+1}\|_F^2 + \frac{\delta}{\beta(1+\delta)}\|\lambda^k-\lambda^{k+1}\|^2 \\ \geq\ & \nu\|u^k-u^{k+1}\|_G^2, \end{aligned} \tag{3.17}$$

where $\nu := \min\left(\delta^2, \frac{\delta}{1+\delta}\right) > 0$, from which the statements of this lemma follow immediately. □

Now, we are ready to prove the convergence of the LALM scheme (3.7).

Theorem 3.3. *For any fixed $\beta>0$ and an arbitrary initial point $\lambda^0\in\mathcal{R}^p$, the sequence $\{(X^k,\lambda^k)\}$ generated by the LALM scheme* (3.7) *with $0<\tau<1/\rho(\mathcal{A}^*\mathcal{A})$ converges to (X^*,λ^*), where X^* is a solution of* (1.1).

Proof. From (a) of Lemma 3.2, it holds that

$$X^k - X^{k+1}\to 0 \quad\text{and}\quad \lambda^k-\lambda^{k+1}\to 0.$$

Thus, it follows from $\lambda^k = \lambda^{k-1}+\beta(b-\mathcal{A}X^k)$ that $\mathcal{A}X^k\to b$. From (b) of Lemma 3.2, $\{u^k\}$ has a subsequence $\{u^{k_j}\}$ converging to $u^\star=(X^\star;\lambda^\star)$, i.e., $X^{k_j}\to X^\star$ and $\lambda^{k_j}\to\lambda^\star$. Next we show that $(X^\star,\lambda^\star)$ satisfies the conditions in (3.8).

First, it follows from $X^{k_j}\to X^\star$ and $\mathcal{A}X^k\to b$ that $\mathcal{A}X^\star = \lim_{j\to\infty}\mathcal{A}X^{k_j} = b$. Second, (3.12) can be rewritten as

$$\frac{\beta}{\tau}(I-\tau\mathcal{A}^*\mathcal{A})(X^k-X^{k+1}) + \mathcal{A}^*\lambda^k + \beta\mathcal{A}^*(b-\mathcal{A}X^{k+1}) \in \partial\|X^{k+1}\|_*. \tag{3.18}$$

Since $X^{k_j} \to X^\star$ and $X^{k+1} - X^k \to 0$, we have

$$X^{k_j+1} = X^{k_j} + (X^{k_j+1} - X^{k_j}) \to X^\star.$$

By taking the limit of (3.18) over k_j and further considering $\mathcal{A}X^k - b \to 0$, it follows that $\mathcal{A}^*\lambda^\star \in \partial\|X^\star\|_*$, which together with $\mathcal{A}X^\star = b$ imply that $(X^\star, \lambda^\star)$ satisfies the optimality conditions in (3.8). Therefore, we have shown that any limit point of $\{(X^k, \lambda^k)\}$ satisfies the conditions in (3.8).

Since (3.17) holds for any (X^*, λ^*) satisfying (3.8), by letting $u^* = (X^*, \lambda^*) = (X^\star, \lambda^\star)$ at the beginning and considering (c) of Lemma 3.2, we obtain the convergence of $\{u^k\}$. □

4. Linearized ADMs for (1.2) and (1.3)

In this section, we extend the linearization technique proposed in Section 3 to solving (1.2) and (1.3). First, we show that the ADM [17] is applicable to some easy reformulations of (1.2) and (1.3). Then, we derive linearized ADMs (LADMs for short) for solving (1.2) and (1.3), and establish their convergence. Due to the obvious similarity, we only elaborate on the details of the LADM for (1.2) and omit those for (1.3).

4.1. LADM for (1.2).

Clearly, by introducing an auxiliary variable $r \in \mathcal{R}^p$, the problem (1.2) can be equivalently transformed to

$$\min_{X\in\mathcal{R}^{m\times n}, r\in\mathcal{R}^p} \left\{ \|X\|_* + \frac{1}{2\mu}\|r\|^2 : \mathcal{A}X - b = r \right\}. \tag{4.1}$$

An obvious advantage of this reformulation is that the objective function of (4.1) has a separable structure, and the ADM [17] is applicable. More specifically, the augmented Lagrangian function of (4.1) is given by

$$\mathcal{L}_U(X, r, \lambda, \beta) := \|X\|_* + \frac{1}{2\mu}\|r\|^2 - \langle \lambda, \mathcal{A}X - r - b\rangle + \frac{\beta}{2}\|\mathcal{A}X - r - b\|^2, \tag{4.2}$$

where $\lambda \in \mathcal{R}^p$ and $\beta > 0$ are defined as before. Given X^k and λ^k, the application of the ADM for (4.1) results in the following iterative scheme:

$$r^{k+1} = \arg\min_r \mathcal{L}_U(X^k, r, \lambda^k, \beta), \tag{4.3a}$$

$$X^{k+1} = \arg\min_X \mathcal{L}_U(X, r^{k+1}, \lambda^k, \beta), \tag{4.3b}$$

$$\lambda^{k+1} = \lambda^k - \beta(\mathcal{A}X^{k+1} - r^{k+1} - b). \tag{4.3c}$$

Therefore, to solve (4.1) by the ADM scheme (4.3), the main computation of each iteration consists of solving two subproblems. More specifically, first it is easy to see that the r-subproblem (4.3a) has the closed-form solution given by

$$r^{k+1} = \frac{\beta\mu}{1+\beta\mu}(\mathcal{A}X^k - b - \lambda^k/\beta).$$

On the other hand, the X-subproblem (4.3b) is equivalent to

$$\min_X \|X\|_* + \frac{\beta}{2}\|\mathcal{A}X - b^k\|^2, \tag{4.4}$$

where $b^k := b + r^{k+1} + \lambda^k/\beta$.

Since (4.4) does not have a closed-form solution for a generic linear operator $\mathcal{A}$, we apply a similar linearization technique in Section 3 to tackle this difficulty.

More specifically, by applying the technique in (3.4), we obtain the following approximated problem to (4.4):

$$\min_X \|X\|_* + \frac{\beta}{2\tau}\|X - (X^k - \tau g^k)\|_F^2, \tag{4.5}$$

where $\tau > 0$ and g^k is the gradient of $\frac{1}{2}\|\mathcal{A}X - b^k\|^2$ at X^k, which is given by

$$g^k := \mathcal{A}^*(\mathcal{A}X^k - b^k) = \mathcal{A}^*(\mathcal{A}X^k - r^{k+1} - b - \lambda^k/\beta). \tag{4.6}$$

The closed-form solution of (4.5) is then readily obtainable by (2.7).

In summary, with the given X^k and λ^k, the proposed LADM for (1.2) generates the next iterate $(r^{k+1}, X^{k+1}, \lambda^{k+1})$ as follows:

$$r^{k+1} = \frac{\beta\mu}{1+\beta\mu}(\mathcal{A}X^k - b - \lambda^k/\beta), \tag{4.7a}$$

$$X^{k+1} = \mathcal{D}_{\tau/\beta}\left(X^k - \tau g^k\right), \tag{4.7b}$$

$$\lambda^{k+1} = \lambda^k - \beta\left(\mathcal{A}X^{k+1} - r^{k+1} - b\right), \tag{4.7c}$$

where g^k is defined in (4.6) and (4.7b) is the solution of (4.5).

4.2. **Convergence analysis.** In this subsection, we establish the global convergence of the LADM scheme (4.7).

Let (r^*, X^*) be any solution of (4.1). From standard theory of convex programming, there exists $\lambda^* \in \mathcal{R}^p$ such that the following conditions are satisfied:

$$r^*/\mu + \lambda^* = 0,\ \mathcal{A}^*\lambda^* \in \partial\|X^*\|_* \ \text{ and } \ \mathcal{A}X^* - b = r^*. \tag{4.8}$$

We first prove a lemma similar to Lemma 3.1 before establishing the convergence.

Lemma 4.1. *Let (r^*, X^*) be an arbitrary solution of (4.1) and $\lambda^* \in \mathcal{R}^p$ be such that the conditions in (4.8) are satisfied. For any fixed $\beta > 0$ and $\tau > 0$ and an arbitrary initial iterate (X^0, λ^0), the sequence $\{(r^k, X^k, \lambda^k)\}$ generated by the LADM scheme (4.7) satisfies*

$$\begin{aligned} \frac{1}{\beta}(\lambda^{k+1} - \lambda^*)^\top(\lambda^k - \lambda^{k+1}) &+ \frac{\beta}{\tau}\langle X^{k+1} - X^*, X^k - X^{k+1}\rangle \\ &\geq (\lambda^k - \lambda^{k+1})^\top\mathcal{A}(X^k - X^{k+1}). \end{aligned} \tag{4.9}$$

Proof. Since r^{k+1} minimizes $\mathcal{L}_U(X^k, r, \lambda^k, \beta)$, it holds that

$$r^{k+1}/\mu + \lambda^k - \beta(\mathcal{A}X^k - r^{k+1} - b) = 0,$$

which, by considering (4.7c), can be rewritten as

$$r^{k+1}/\mu + \lambda^{k+1} + \beta\mathcal{A}(X^{k+1} - X^k) = 0.$$

Further, considering $r^*/\mu + \lambda^* = 0$, we obtain

$$(r^{k+1} - r^*)/\mu = \beta\mathcal{A}(X^k - X^{k+1}) - (\lambda^{k+1} - \lambda^*).$$

Therefore, it holds that

$$(r^{k+1} - r^*)^\top\left(\beta\mathcal{A}(X^k - X^{k+1}) - (\lambda^{k+1} - \lambda^*)\right) = \|r^{k+1} - r^*\|^2/\mu \geq 0. \tag{4.10}$$

Similarly, the optimality condition of (4.5) implies that

$$0 \in \partial\|X^{k+1}\|_* + \frac{\beta}{\tau}(X^{k+1} - X^k + \tau g^k), \tag{4.11}$$

where g^k is defined in (4.6). Plugging (4.6) into (4.11) and considering (4.7c), we can rewrite (4.11) as

$$\frac{\beta}{\tau}\left(I-\tau\mathcal{A}^*\mathcal{A}\right)\left(X^k-X^{k+1}\right)+\mathcal{A}^*\lambda^{k+1}\in\partial\|X^{k+1}\|_*. \tag{4.12}$$

Further considering $\mathcal{A}^*\lambda^*\in\partial\|X^*\|_*$ and the convexity of $\|\cdot\|_*$, there holds

$$\left\langle X^{k+1}-X^*,\frac{\beta}{\tau}\left(I-\tau\mathcal{A}^*\mathcal{A}\right)\left(X^k-X^{k+1}\right)+\mathcal{A}^*(\lambda^{k+1}-\lambda^*)\right\rangle\geq 0,$$

which is equivalent to

$$\begin{aligned}&\left\langle\mathcal{A}(X^*-X^{k+1}),\beta\mathcal{A}(X^k-X^{k+1})-(\lambda^{k+1}-\lambda^*)\right\rangle\\&\quad+\frac{\beta}{\tau}\left\langle X^{k+1}-X^*,X^k-X^{k+1}\right\rangle\geq 0.\end{aligned} \tag{4.13}$$

It follows from $\mathcal{A}X^*-r^*=b$ and $\beta(\mathcal{A}X^{k+1}-r^{k+1}-b)=\lambda^k-\lambda^{k+1}$ that the addition of (4.10) and (4.13) gives rise to (4.9) immediately. □

Let (r^*,X^*,λ^*) be arbitrarily chosen such that the conditions in (4.8) are satisfied and assume that $0<\tau<1/\rho(\mathcal{A}^*\mathcal{A})$. Using the same notation defined in (3.14) and exactly the same arguments presented in the proof of Lemma 3.2, we can show that, for any fixed $\beta>0$ and an arbitrary initial iterate (X^0,λ^0), the sequence $\{(r^k,X^k,\lambda^k)\}$ generated by the LADM scheme (4.7) satisfies

$$\|u^k-u^*\|_G^2-\|u^{k+1}-u^*\|_G^2\geq\nu\|u^k-u^{k+1}\|_G^2, \tag{4.14}$$

for some $\nu>0$. Therefore, the three conditions (a), (b) and (c) in Lemma 3.2 are also satisfied.

Now, we are ready to prove the convergence of the LADM scheme (4.7).

Theorem 4.2. *For any fixed $\beta>0$ and an arbitrary initial iterate (X^0,λ^0), the sequence $\{(r^k,X^k,\lambda^k)\}$ generated by the LADM scheme (4.7) with $0<\tau<1/\rho(\mathcal{A}^*\mathcal{A})$ converges to (r^*,X^*,λ^*), where (r^*,X^*) is a solution of (4.1).*

Proof. It follows from $\|u^k-u^{k+1}\|_G\to 0$ that

$$X^k-X^{k+1}\to 0\quad\text{and}\quad\lambda^k-\lambda^{k+1}\to 0.$$

Further, considering $\lambda^k=\lambda^{k-1}-\beta(\mathcal{A}X^k-r^k-b)$, we obtain

$$\mathcal{A}X^k-r^k-b\to 0.$$

Since $\{u^k\}$ lies in a compact region, it has a subsequence $\{u^{k_j}\}$ converging to $u^\star=(X^\star;\lambda^\star)$, i.e., $X^{k_j}\to X^\star$ and $\lambda^{k_j}\to\lambda^\star$. In addition, (4.7a) implies that

$$r^k=\frac{\beta\mu}{1+\beta\mu}\left(\mathcal{A}X^k-b-\lambda^k/\beta+\mathcal{A}(X^{k-1}-X^k)+(\lambda^k-\lambda^{k-1})/\beta\right).$$

Considering $X^{k_j}\to X^\star$, $\lambda^{k_j}\to\lambda^\star$, $X^{k-1}-X^k\to 0$ and $\lambda^{k-1}-\lambda^k\to 0$, the above equality implies that

$$r^{k_j}\to r^\star:=\frac{\beta\mu}{1+\beta\mu}\left(\mathcal{A}X^\star-b-\lambda^\star/\beta\right),\quad j\to\infty. \tag{4.15}$$

Therefore, $(r^\star,X^\star,\lambda^\star)$ is also a limit point of $\{(r^k,X^k,\lambda^k)\}$.

Next we show that $(r^\star, X^\star, \lambda^\star)$ satisfies the optimality conditions in (4.8). First, from (4.7) we have

$$\lambda^{k+1} = \lambda^k - \beta\left(\mathcal{A}X^{k+1} - \frac{\beta\mu}{1+\beta\mu}\left(\mathcal{A}X^k - b - \lambda^k/\beta\right) - b\right),$$

which is easily shown to be equivalent to

$$\frac{\lambda^k - \lambda^{k+1}}{\beta} + \mathcal{A}(X^k - X^{k+1}) = \frac{1}{1+\beta\mu}\left(\mathcal{A}X^k - b + \mu\lambda^k\right).$$

By taking the limit of the above equality over k_j, it follows that

$$\mathcal{A}X^\star - b + \mu\lambda^\star = 0. \tag{4.16}$$

Second, from the definition of $r^\star$ in (4.15), it is easy to verify that

$$r^\star/\mu + \lambda^\star = \frac{\beta}{1+\beta\mu}\left(\mathcal{A}X^\star - b + \mu\lambda^\star\right) = 0, \tag{4.17}$$

where the second equality follows from (4.16). Finally, (4.12) can be rewritten as

$$\frac{\beta}{\tau}(I - \tau\mathcal{A}^*\mathcal{A})(X^k - X^{k+1}) - \beta\mathcal{A}^*(\mathcal{A}X^{k+1} - r^{k+1} - b) + \mathcal{A}^*\lambda^k \in \partial\|X^{k+1}\|_*. \tag{4.18}$$

Since $X^{k_j} \to X^\star$ and $X^{k+1} - X^k \to 0$, we have

$$X^{k_j+1} = X^{k_j} + (X^{k_j+1} - X^{k_j}) \to X^\star.$$

By taking the limit of (4.18) over k_j and further considering $\mathcal{A}X^k - r^k - b \to 0$, it follows that $\mathcal{A}^*\lambda^\star \in \partial\|X^\star\|_*$, which together with equations (4.16) and (4.17) imply that $(r^\star, X^\star, \lambda^\star)$ satisfies the optimality conditions (4.8). Therefore, we have shown that any limit point of $\{(r^k, X^k, \lambda^k)\}$ is an optimal solution of (4.1).

Since (4.14) holds for any optimal solution of (4.1), by letting $u^* = (X^*, \lambda^*) = (X^\star, \lambda^\star)$ at the beginning and considering (c) of Lemma 3.2, we obtain the convergence of $\{u^k\}$, and thus that of $\{(r^k, X^k, \lambda^k)\}$. □

4.3. **LADM for** (1.3)**.** Analogously, a LADM for (1.3) can also be easily derived. Note that (1.3) is equivalent to

$$\min_{X\in\mathcal{R}^{m\times n}, r\in\mathcal{R}^p}\left\{\|X\|_* : \mathcal{A}X - b = r \in B_{2,\delta}\right\}, \tag{4.19}$$

where $r \in \mathcal{R}^p$ is an auxiliary variable and $B_{2,\delta} := \{\xi \in \mathcal{R}^p : \|\xi\| \le \delta\}$. The augmented Lagrangian function of (4.19) is given by

$$\mathcal{L}_C(X, r, \lambda, \beta) := \|X\|_* - \langle\lambda, \mathcal{A}X - r - b\rangle + \frac{\beta}{2}\|\mathcal{A}X - r - b\|^2.$$

The ADM scheme for (4.19) is exactly the same form as in (4.3) except that $\mathcal{L}_U$ is replaced by $\mathcal{L}_C$ and the minimization for r is over $B_{2,\delta}$. It is easy to show that the minimization of $\mathcal{L}_C(X^k, r, \lambda^k, \beta)$ over $B_{2,\delta}$ is given by $r^{k+1} = \mathcal{P}_{B_{2,\delta}}\left(\mathcal{A}X^k - b - \lambda^k/\beta\right)$. By applying the approximation technique (3.4) to $\mathcal{L}_C(X, r^{k+1}, \lambda^k, \beta)$, the LADM for (4.19) is as follows:

$$r^{k+1} = \mathcal{P}_{B_{2,\delta}}\left(\mathcal{A}X^k - b - \lambda^k/\beta\right), \tag{4.20a}$$

$$X^{k+1} = \mathcal{D}_{\tau/\beta}\left(X^k - \tau g^k\right), \tag{4.20b}$$

$$\lambda^{k+1} = \lambda^k - \beta(\mathcal{A}X^{k+1} - r^{k+1} - b), \tag{4.20c}$$

where g^k is defined in (4.6). Clearly, (4.20) differs with (4.7) only in the iteration for r. The convergence of (4.20) to a solution of (4.19) is summarized in the following theorem, whose proof is similar to that of Theorem 4.2 and thus is omitted.

Theorem 4.3. *For any fixed $\beta > 0$ and an arbitrary initial iterate (X^0, λ^0), the sequences $\{(r^k, X^k, \lambda^k)\}$ generated by the LADM scheme (4.20) with $0 < \tau < 1/\rho(\mathcal{A}^*\mathcal{A})$ converges to $\{(r^*, X^*, \lambda^*)\}$, where (r^*, X^*) is a solution of (4.19).*

5. Connections to existing work

In this section, we elucidate the connections between the proposed linearized methods and some existing approaches in the literature for linear inverse problems, including total variation problems in image restoration, ℓ_1-problems in compressive sensing, and nuclear norm related problems in matrix completion.

5.1. **Connections to proximal forward-backward operator splitting methods.** In fact, the blend of the linearization and proximal techniques (3.4) can be viewed as a proximal forward-backward operator splitting method, which has been applied to various inverse problems in the literature; see, e.g., [11] and references therein. Specifically, based on the forward-backward operator splitting a fixed-point algorithm is derived in [20] for the unconstrained ℓ_1-problem in compressive sensing:

$$\min_{x\in\mathcal{R}^n} \|x\|_1 + \frac{1}{2\mu}\|Ax-b\|^2, \tag{5.1}$$

where $A \in \mathcal{R}^{m\times n}$ and $b \in \mathcal{R}^m$. The iterative scheme in [20] is

$$y^k = x^k - \tau g^k, \tag{5.2a}$$

$$x^{k+1} = \max(|y^k| - \tau\mu, 0) \circ \operatorname{sgn}(y^k), \tag{5.2b}$$

where $g^k = A^\top(Ax^k - b)$, $\tau > 0$ is a proximal parameter, and $|\cdot|$, sgn and "$\circ$" denote componentwise absolute value, signum and multiplication, respectively. Aided by continuation and line-search techniques, fast convergence of (5.2) is demonstrated in [20]. Subsequently, this approach is extended in [32] to solving (1.2), and its iterative scheme is

$$Y^k = X^k - \tau g^k, \tag{5.3a}$$

$$X^{k+1} = \mathcal{D}_{\tau\mu}(Y^k), \tag{5.3b}$$

where $g^k = \mathcal{A}^*(\mathcal{A}X^k - b)$, $\tau > 0$ is a parameter and $\mathcal{D}_{\tau\mu}$ is defined in (2.6).

It is shown in [20] that the convergence of (5.2) is guaranteed provided that $0 < \tau < 2/\rho(A^\top A)$. Similar results are obtained in [32] for (5.3). In fact, the accelerated proximal point algorithm derived in [49] is also based on the same forward-backward operator splitting idea, besides the acceleration step.

We note that, by Theorems 3.3, 4.2 and 4.3, the convergence of the proposed linearized methods (LALM (3.7), LADMs (4.7) and (4.20)) is guaranteed only for $0 < \tau < 1/\rho(\mathcal{A}^*\mathcal{A})$, which is a narrower interval than those results in [20, 32]. One explanation for this difference is that the linearization technique (3.4) is directly applied to the original problems (5.1) and (1.2) in [20] and [32], respectively; while the proposed linearized methods apply the linearization technique to the resulting subproblems. Despite this difference, our extensive experimental results show that the proposed linearized methods are in general much faster than the accelerated

versions of the fixed-point iterations (5.2) and (5.3) by the continuation and line-search techniques. We refer to [10] and [55] for similar discussions on the matrix completion problems and ℓ_1-problems in compressive sensing, respectively. We believe that the promising convergence of the proposed LALM and LADMs is mainly attributed to the timely update of the Lagrange multiplier right after each round of alternating minimization of the variables.

In Section 6, we will compare the proposed LADM (4.7) numerically with the accelerated proximal gradient algorithm [49], which has been shown to be much faster than the fixed-point continuation approach [32].

5.2. **Connections to Bregman-related algorithms.** In this subsection, we delineate the relationships between the proposed linearized methods and the Bregman related iterative algorithms [37, 57, 58] for linear inverse problems.

In fact, it is shown in [57] that the ALM for equality constrained problems is equivalent to the Bregman iterative algorithm originally proposed in [37]. Therefore, the ALM scheme (3.2) is equivalent to the Bregman iterative algorithm applied to (1.1). In [32], the authors propose to use the Bregman iterative algorithm for solving (1.1), resulting in an iterative framework in the form (3.3) (see [32, Eq. (5.10)]), where the X-subproblem (3.3a) is solved iteratively by the fixed-point scheme (5.3). An obvious advantage of the LALM scheme (3.7) is that there is no need to solve any subproblem iteratively.

The LALM scheme (3.7) is also closely related to the linearized Bregman method proposed in [57]. Let

$$v^k := \mathcal{A}^*\lambda^k/\beta,\ \ Y^k := X^k - \tau\mathcal{A}^*\left(\mathcal{A}X^k - b\right),\ \text{ and }\ J(X) := \frac{1}{\beta}\|X\|_*.$$

By noting (3.6), we can rewrite the LALM scheme (3.7) as

$$X^{k+1} = \arg\min_X J(X) + \frac{1}{2\tau}\|X - (Y^k + \tau v^k)\|_F^2, \tag{5.4a}$$

$$v^{k+1} = v^k + \mathcal{A}^*\left(b - \mathcal{A}X^{k+1}\right). \tag{5.4b}$$

While, the linearized Bregman method for (1.1) (see [57, Eqs. (5.15) and (5.17)] and [56, Eqs. (1.9)-(1.10)]) reads

$$X^{k+1} = \arg\min_X J(X) + \frac{1}{2\tau}\|X - (Y^k + \tau v^k)\|_F^2, \tag{5.5a}$$

$$v^{k+1} = v^k - \mathcal{A}^*\left(\mathcal{A}X^k - b\right) - (X^{k+1} - X^k)/\tau. \tag{5.5b}$$

On the other hand, by a change of variables $v^k \leftarrow v^k - \mathcal{A}^*(\mathcal{A}X^k - b) + X^k/\tau$ in (5.5), we obtain another form of the linearized Bregman method (see [56, Eq. (1.13)]):

$$X^{k+1} = \arg\min_X J(X) + \frac{1}{2\tau}\|X - \tau v^k\|_F^2, \tag{5.6a}$$

$$v^{k+1} = v^k + \mathcal{A}^*\left(b - \mathcal{A}X^{k+1}\right). \tag{5.6b}$$

Note that (5.5) is actually the original form of the linearized Bregman method proposed in [57]. By comparing (5.4) with (5.5) and (5.6), we see the difference of these two method clearly. Moreover, we emphasize that the proposed LALM differs from the linearized Bregman method essentially in the convergence aspect: the former converges to a solution of the original problem (1.1), while the latter

converges to the solution of the problem

$$\min_X \left\{ J(X) + \frac{1}{2\tau}\|X\|_F^2 : \mathcal{A}X = b \right\}, \tag{5.7}$$

which is only an approximation to (1.1). Note that the problem (5.7) approximates to the original problem (1.1) well only when τ is large (for fixed β). We refer to [5, 6] for analogous discussions on the basis pursuit problem.

After the first round of referee review, our attention was brought to the Bregmanized operator splitting (BOS) method proposed in [58] for linear equality constrained problems in the form of (1.1), where a generic convex objective function is studied in place of the nuclear norm. By a change of variable $b^k := \lambda^k/\beta + b$, the LALM scheme (3.7) can be rewritten as

$$X^{k+1} = \mathcal{D}_{\tau/\beta}\left(X^k - \tau\mathcal{A}^*(\mathcal{A}X^k - b^k)\right), \tag{5.8a}$$

$$b^{k+1} = b^k + b - \mathcal{A}X^{k+1}, \tag{5.8b}$$

which is actually equivalent to the BOS method (see [58, **Algorithm I**]). Therefore, we have obtained the same method from different motivations: our idea is to linearize the augmented Lagrangian function (3.1) in the ALM framework (3.2), while the BOS method is derived from the Bregman iterative framework [37] with operator splitting. The motivation of linearizing the augmented Lagrangian function is natural. More importantly, our idea can be easily extended to solve (1.2) and (1.3), while the BOS method is only derived for equality constrained problems. Besides, we have obtained stronger convergence results under the same condition as in [58]. In fact, it is proved in [58, Theorem 1] that any accumulation point of the sequence generated by the BOS method is a solution of (1.1), while we establish respectively the global convergence of the LALM and LADMs to solutions of (1.1)-(1.3).

We note that the LADMs (4.7) and (4.20) are also related to the idea of the BOS method in the following sense. In fact, the iteration of primal variables in the BOS method is a gradient descent step followed by a shrinkage step (or a generalized projection step in the sense of Moreau [41, Theorem 31.5]). Similarly, since the Hessian of $\mathcal{L}_U(X^k, r, \lambda^k, \beta)$ with respect to r is identity, the iteration (4.7a) can be viewed as a steepest gradient descent step. On the other hand, the iteration of X in (4.7b) is a gradient descent step (with the latest updated data) followed by a shrinkage step. As such, (4.7a)-(4.7b) is also a generalized projected gradient descent step. The iteration of dual variables in both BOS and LADMs is in fact identical.

6. Numerical results

In this section, we compare the proposed LALM and LADMs numerically with some existing efficient methods and report the numerical results. In particular, we compare the proposed methods with the accelerated proximal gradient algorithm with line-search (APGL) in [49] and the primal algorithm in [38] due to their significant superiorities to many other existing methods; see the extensive numerical results in [38, 49] for matrix completion problems, gene expression examples, and others. We also compare the proposed linearized methods with the ADMs proposed in [10] for matrix completion problems and those discussed in Section 2 for (1.1) and (1.3) with $\mathcal{A}\mathcal{A}^* = \mathcal{I}$.

All the experiments were performed under Windows Vista Premium and Matlab v7.10 (R2010a), running on a Lenovo laptop with an Intel Core 2 Duo CPU at 1.8 GHz and 2GB of memory.

We note that the APGL code[1] is applicable to the case with a generic linear operator $\mathcal{A}$, while it can only solve the unconstrained model (1.2) (an additional regularization of the form $\|X\|_F^2$ is also allowed). Moreover, the primal and dual algorithms [38] (implemented in [30]) are especially designed for an unconstrained model of the following form

$$\min_{X\in\mathcal{R}^{m\times n}} \|X\|_* + \frac{1}{2\mu}\|AX - B\|_F^2, \tag{6.1}$$

where $\mu > 0$, $A \in \mathcal{R}^{p\times m}$, $B \in \mathcal{R}^{p\times n}$, and AX represents the ordinary matrix multiplication. In [38], the authors first reformulate (6.1) into one that has exactly the same form as itself, where A, B and X are replaced by certain $\tilde{A}$, $\tilde{B}$ and $\tilde{X}$, respectively, with $\tilde{A}$ being full column rank (see [38, Proposition 1]). Then the authors study gradient-projection methods for the dual problem. In addition, Nesterov's accelerated first-order methods [33, 34] are also applied to solve the reduced problem of (6.1), where a practical stopping criterion is derived based on both the primal and the dual problems. Therefore, the primal method in [38] is essentially a variant of the accelerated proximal gradient method, and it shares the same idea of using the linearization and proximal techniques as the proposed methods. Due to this similarity, we only compare with the primal method in [38] (implemented in the SLEP package as "`mat_primal`"). We refer to [38] for extensive numerical results on multi-task learning problems for the relative performance of the primal and the dual methods derived therein.

For a matrix M, we let $\mathtt{vec}(M)$ be the vector stagnated by the columns of M. Then, (6.1) is a special case of (1.2) with

$$\mathcal{A}X = \mathtt{vec}(AX) = \mathrm{diag}(\overbrace{A,\ldots,A}^{n \text{ times}})\mathtt{vec}(X) \ \text{ and } \ b = \mathtt{vec}(B). \tag{6.2}$$

On the contrary, with the given $\mathcal{A}$ and b, in general one cannot find matrices A and B such that (6.1) is equivalent to (1.2). For example, the matrix completion problem, where $\mathcal{A}$ is defined in (1.4), cannot be written in the form of (6.1) in general, and thus the algorithms in [38] are not applicable. In comparison, the proposed LALM and LADMs are easily applicable to solve all these models (1.1)-(1.3). If $\mathcal{A}\mathcal{A}^* = \mathcal{I}$, as we show in Section 2, the ADM is applicable. In short, the algorithms proposed in this paper have much larger applicable range than those in the literature.

6.1. **Experiments and implementation details.** Our numerical experiments are categorized into the following three classes.

(1) Compare the LALM (3.7) and the LADM (4.20) with the ADMs in [10] on matrix completion problems. We concentrate on the constrained models (1.1) and (1.3).
(2) Compare the LALM (3.7) and the LADM (4.20) with the ADMs discussed in Section 2 for solving the constrained models (1.1) and (1.3), where $\mathcal{A}$ is a two-dimensional partial DCT (discrete cosine transform) operator.

[1] Available at: `http://www.math.nus.edu.sg/~mattohkc/NNLS.html`

(3) Compare the LADM (4.7) with the APGL [49] and the primal algorithm in [38] for solving (6.1) with random data.

In the first two classes of experiments we do not compare with the APGL on the unconstrained model (1.2) for reasons given below. First, the relative performance of the ADM compared with APGL has been well illustrated in [10] on matrix completion problems with both random and gene expression data. Roughly, to generate solutions of the same quality, the ADM is faster than the APGL on matrix completion problems whenever the sample ratio (denoted by `sr` and defined by $|\Omega|/mn$) is relatively high. But, we also note that the APGL performs better than the ADM when n is large and meanwhile the sample ratio is small (say, $n > 5000$ and $sr \leq 10\%$). Second, the relative performance between the LADM (4.7) and the APGL on the problems tested in the first two classes of experiments can be roughly estimated based on both results in [10] and those to be presented in subsections 6.2 and 6.3. Furthermore, the LADM (4.7) is compared with the APGL in the third class of experiments for solving (6.1) with various of random data.

In all experiments, we generated X^* via the Matlab script

$$\text{“}\texttt{randn}(m, r) * \texttt{randn}(r, n)\text{”},$$

where r is a prefixed integer. For matrix completion problems, we generated the index set Ω randomly. For the second class of experiments, the partial DCT operator was also generated randomly. Then, we set $b = \mathcal{A}X^* + \omega$, where ω is a white noise of mean zero and standard deviation `std`.

Next, we clarify some details for implementing the proposed algorithms.

- Partial SVD. As we pointed out before, each iteration of the proposed LALM and LADMs is dominated by one SVD, as conventionally required by most of the existing methods for nuclear norm related problems. In our implementation, we employ the influential PROPACK package [28] to realize partial SVD for all the proposed methods and the ADMs. In particular, we only need those singular values bigger than a threshold and the corresponding singular vectors. However, it is well known that PROPACK is not able to automatically compute those singular values bigger than a prefixed threshold, but only able to compute a prefixed number of them. Therefore, we need to efficiently determine the number of singular values to be computed at each iteration empirically.

  Let sv^k denote the number of singular values to be computed at the k-th iteration, whose initial value is given by $sv^0 = \min(m, n)/20$. We use the same strategy as in [49] to update sv^k, that is,

$$sv^{k+1} = \begin{cases} svp^k + 1, & \text{if } svp^k < sv^k, \\ svp^k + 5, & \text{if } svp^k = sv^k, \end{cases}$$

  where svp^k represents the number of positive singular values of X^k. Based on our experiments, this adjusting rule works very well: after a few iterations, the rank of X^* can be well estimated.
- The penalty parameter β. As we have proved, the proposed LALM and LADMs all converge globally for any fixed $\beta > 0$. However, different choice of β affects the effectiveness of the proposed LALM and LADMs. In our experiment, we set $\beta = 2.5/\min(m, n)$ empirically, which works quite well for the tested problems. We will present some experimental results to

illustrate how the numerical performance of the proposed algorithms is affected by different values of β.

- The proximal parameter τ. The proximal parameter τ plays also an important role for the effectiveness of the proposed methods. Theoretically, we show the convergence of the proposed methods under the condition $0 < \tau < 1/\rho(\mathcal{A}^*\mathcal{A})$. But empirically, we find that values of τ slightly greater than $1/\rho(\mathcal{A}^*\mathcal{A})$ can accelerate the convergence. For this reason, in all our experiments, we set $\tau = 1/\rho(\mathcal{A}^*\mathcal{A})$. Note that for matrix completion problems and partial DCT measurements, $\rho(\mathcal{A}^*\mathcal{A}) = 1$; while for random data we compute $\rho(\mathcal{A}^*\mathcal{A})$ in advance for once.

For the ADMs in [10] and those discussed in Section 2, we also employed the aforementioned implementation techniques for the proposed LALM and LADMs, including the partial SVD and the choice of the penalty parameter, etc. Moreover, we terminated ADMs, LALM and LADMs by the following criterion:

$$\text{RelChg} = \frac{\|X^k - X^{k-1}\|_F}{\max(\|X^{k-1}\|_F, 1)} \leq tol,$$

where $tol > 0$ is a given tolerance. As discussed in [55] for ℓ_1-problems, solving optimization problems with very high accuracy does not necessarily result in high quality solutions for noisy data. Thus, we set $tol = 10^{-5}$ for the noiseless model (1.1) and $tol = 10^{-4}$ for the noisy models (1.2) and (1.3) in all of the experiments. For APGL and `mat_primal`, we terminated both algorithms by setting the value of tolerance to be 10^{-4}. That is, we set `par.tol` $= 10^{-4}$ in APGL and `opt.tol` $= 10^{-4}$ in `mat_primal`. Other algorithmic parameters in APGL and `mat_primal` are set to their default values.

6.2. **Comparison results of LALM, LADMs and ADMs: matrix completion via** (1.1) **and** (1.3)**.** In this subsection, we compare the proposed LALM and LADMs with the ADMs [10] on matrix completion problems. We set the noise level `std` $= 0.001$, $\delta = \|\omega\|$ in (1.3) and $m = n$ in all of the tests.

In the results presented below, r, `sr`, p and `dof` denote, respectively, the rank of X^*, sample ratios taken, the number of measurements and the "degree of freedom" defined by $r(m + n - r)$ for a matrix with rank r. The number of iterations and consumed CPU time (measured in seconds) are denoted by "iter" and "CPU", respectively. We measure the quality of a recovered solution by its relative error to the true low-rank matrix X^*, which is denoted by "RErr" and is defined by

$$\text{RErr} = \|X^k - X^*\|_F / \|X^*\|_F.$$

For each scenario, we generated the model by 10 times and reported the average results. The results for noiseless and noisy data are presented, respectively, in Tables 1 and 2.

As shown by Tables 1 and 2, the proposed LALM and LADMs both perform very well on the tested matrix completion problems. Specifically, for both noiseless and noisy data, LALM and LADMs can obtain solutions which are of almost equally good quality as those by exact ADMs. Surprisingly, the number of iterations taken by LALM and LADMs are almost equal to those taken by the exact ADMs. Since the per-iteration cost of both algorithms is roughly the same (both are dominated by one SVD), the CPU time consumed by both are also roughly equal. These results clearly demonstrate that the proposed LALM and LADMs can be potentially

TABLE 1. Comparison results of LALM (3.7) and ADM in [10] (the same as (2.11)) on (1.1): matrix completion with noiseless data.

Unknown X			ADM			LALM		
(n,r)	sr	$p/$dof	iter	RErr	CPU	iter	RErr	CPU
(500,10)	60%	15.15	24.9	4.76e-6	5.6	24.8	9.16e-6	5.8
	40%	10.10	39.1	8.44e-6	6.4	38.8	8.14e-6	6.9
	20%	5.05	73.0	1.02e-5	8.5	73.2	1.02e-5	9.3
(1000,20)	60%	15.15	25.4	4.42e-6	19.5	24.3	9.18e-6	16.8
	40%	10.10	38.8	9.32e-6	20.8	38.9	9.04e-6	21.3
	20%	5.05	74.3	9.46e-6	26.1	74.1	9.46e-6	24.9
(2000,30)	60%	20.15	24.7	3.89e-6	56.5	23.9	8.84e-6	54.9
	40%	13.43	41.1	6.71e-6	68.1	41.5	6.69e-6	64.6
	20%	6.72	77.5	8.88e-6	77.6	77.3	8.88e-6	75.9
(5000,50)	20%	10.05	81.9	8.77e-6	637	82.1	8.77e-6	679

TABLE 2. Comparison results of LADM (4.20) and ADM in [10] (the same as (2.11) with $\mathcal{P}_{\mathbf{B}_0}$ replaced by $\mathcal{P}_{\mathbf{B}_\delta}$) on (1.3): matrix completion with noisy data.

Unknown X			ADM			LADM		
(n,r)	sr	$p/$dof	iter	RErr	CPU	iter	RErr	CPU
(500,10)	60%	15.15	19.8	2.74e-4	5.0	20.3	2.87e-4	4.9
	40%	10.10	30.5	4.94e-4	5.4	30.1	4.90e-4	5.5
	20%	5.05	53.1	3.58e-4	7.1	52.6	3.58e-4	6.9
(1000,20)	60%	15.15	20.1	1.73e-4	14.8	19.9	1.85e-4	14.7
	40%	10.10	29.7	3.98e-4	17.4	28.8	3.95e-4	16.2
	20%	5.05	53.3	2.22e-4	20.7	53.1	2.22e-4	20.4
(2000,30)	60%	20.15	19.4	9.00e-5	47.8	18.9	1.10e-4	49.0
	40%	13.43	30.3	4.12e-4	54.5	30.4	4.08e-4	52.7
	20%	6.72	57.7	4.16e-4	64.4	58.2	4.16e-4	63.5
(5000,50)	20%	10.05	62.7	1.24e-4	530	63.0	1.24e-4	532

as efficient as exact ADMs. Since our experiments were completed on a laptop with 2GB RAM, we were not able to test larger scenarios where $m, n > 5000$ and $rank(X^*) > 50$, due to the memory limitation.

6.3. **Comparison results of LALM, LADMs and ADM: partial DCT data.** In this subsection, we compare the proposed LALM and LADMs with the ADM on the partial DCT data. The same as in subsection 6.2, we set $\texttt{std} = 0.001$, $\delta = \|\omega\|$ in (1.3) for noisy data and $m = n$ in all the tests. For each scenario, we generated the model by 10 times and reported the average results in Tables 3 and 4.

Similar to the results for the matrix completion problem, the results in Tables 3 and 4 show that the LALM (3.7) performs almost equally well as the ADM on solving (1.1) and (1.3) with partial DCT data. It can be seen that longer CPU time is consumed by both methods for partial DCT data than the corresponding experiments on matrix completion. This is because the computation of $\mathcal{A}X$ and

TABLE 3. Comparison results of LALM (3.7) and ADM (2.11) on (1.1): noiseless partial DCT data.

Unknown X			ADM			LALM		
(n,r)	sr	$p/$dof	iter	RErr	CPU	iter	RErr	CPU
(500,10)	60%	15.15	25.3	5.59e-6	14.2	24.9	7.03e-6	13.5
	40%	10.10	38.9	8.82e-6	20.6	39.4	8.58e-6	21.4
	20%	5.05	75.3	1.40e-5	39.2	75.4	1.40e-5	39.1
(1000,20)	60%	15.15	25.1	4.51e-6	44.4	24.2	9.27e-6	43.0
	40%	10.10	38.9	9.36e-6	66.9	39.2	9.07e-6	67.5
	20%	5.05	73.7	1.00e-5	123	73.6	1.00e-5	125
(2000,30)	60%	20.15	25.7	4.48e-6	187	24.0	9.03e-6	176
	40%	13.43	41.2	6.77e-6	293	41.5	6.75e-6	295
	20%	6.72	77.2	8.95e-6	537	76.7	8.97e-6	538
(3000,50)	20%	6.05	77.4	9.02e-6	1207	76.3	9.46e-6	1198

TABLE 4. Comparison results of LADM (4.20) and ADM (that is, (2.11) with $\mathcal{P}_{\mathbf{B}_0}$ replaced by $\mathcal{P}_{\mathbf{B}_\delta}$) on (1.3): noisy partial DCT data.

Unknown X			ADM			LADM		
(n,r)	sr	$p/$dof	iter	RErr	CPU	iter	RErr	CPU
(500,10)	60%	15.15	20.3	1.24e-4	12.1	20.5	2.85e-4	11.9
	40%	10.10	31.5	1.65e-4	16.1	29.9	4.93e-4	15.8
	20%	5.05	54.0	2.80e-4	28.1	53.4	3.92e-4	28.0
(1000,20)	60%	15.15	20.2	1.03e-4	35.1	20.4	1.85e-4	35.0
	40%	10.10	29.8	1.85e-4	51.4	30.2	3.95e-4	50.3
	20%	5.05	54.3	2.45e-4	90.0	53.7	2.23e-4	84.9
(2000,30)	60%	20.15	19.4	1.69e-4	125	19.3	1.09e-4	123
	40%	13.43	32.8	1.95e-4	212	30.9	4.13e-4	185
	20%	6.72	59.4	2.10e-4	369	58.3	4.17e-4	345
(3000,50)	20%	6.05	58.5	1.42e-4	918	57.8	3.08e-4	901

$\mathcal{A}^*y$ is no longer trivial for a generic linear operator $\mathcal{A}$. For partial DCT data, we were only able to test problems as large as $m = n = 3000$, $rank(X^*) = 50$ and sr $= 20\%$, which is smaller than the limitation scale for matrix completion as presented in subsection 6.2. This is because for matrix completion problems the explicit storage of the matrix variable X can be avoided during the whole iteration process due to the speciality of the linear operator $\mathcal{A}$ defined in (1.4). In fact, in the PROPACK package modified by the authors of [49], X is stored by its UV-decomposition for matrix completion problems, i.e., $X = UV^\top$, where U and V are tall and thin matrices whose number of columns is dynamically adjusted. In comparison, for partial DCT operator, X needs to be stored explicitly in order to compute $\mathcal{A}X$.

It is also easy to see from Tables 1-4 that the iteration numbers consumed by the ADMs, LALM and LADMs all increase moderately as sample ratios become lower. An explanation is that, when the number of constraints becomes fewer, the two variable blocks in the ADM framework have more freedom in the whole space,

and hence ADM, as a blockwise alternating coordinate descent method, can be less efficient. This also explains why ADMs are less competitive to the APGL when sample ratio is relatively low, as shown in [10].

6.4. **Comparison results of LADM, APGL and `mat_primal`: solving (6.1) with random data.** In this subsection, we compare the proposed LADMs (4.7) with APGL [49] and `mat_primal` [38] on solving (6.1) with randomly generated data. The data matrix A is generated by the following Matlab scripts.

```
(1) sqrt_lam = sqrt(lam_max);
(2) [P,~ ] = qr(randn(p));
(3) d = [1; 1 + rand(min(p,m)-2,1)*(sqrt_lam-1); sqrt_lam];
(4) [Q,~ ] = qr(randn(m));
(5) A = P * spdiags(d,0,p,m) * Q'.
```

Here `lam_max` is a prefixed number to control the maximum eigenvalue of $AA^\top$. In fact, for A generated by the above Matlab scripts, the minimum and maximum eigenvalues of $AA^\top$ are, respectively, 1 and `lam_max`, while the others are randomly distributed in [1, `lam_max`]. For $\mathcal{A}$ given in (6.2), it holds that $\rho(\mathcal{A}^*\mathcal{A}) = \rho(A^\top A)$. In all experiments, we set $\tau = 1/\rho(\mathcal{A}^*\mathcal{A}) = 1/$`lam_max`. The low-rank matrix X^* and B are then generated by $X^* =$ `randn`$(m, r) *$ `randn`(r, n) and $B = AX^* +$ `std` $*$ `randn`(p, n), respectively, where `std` $= 0.001$. In our experiments, we tested different combinations of `lam_max` and μ: (`lam_max`$, \mu) \in \{10^0, 10^1, 10^2\} \times \{10^{-2}, 10^0, 10^2\}$.

We note that in general the purpose of solving (6.1) with the data given by "$B = AX^* +$ `Noise`" is not to recover X^* with high accuracy (especially when the observed data is insufficient), but to determine a trade-off solution between the low-rank and data-fidelity purposes. On the other hand, for unconstrained optimization problems the objective function values illustrate the quality of solutions well from the optimization point of view. Therefore, in the following, instead of the relative errors to X^*, we report the resulting objective function values to illustrate the performance of the compared algorithms. We also refer to [38], where the quality of computed solutions to (6.1) is measured by the duality gap.

The comparison results of APGL, `mat_primal` and LADM (4.7) are presented in Table 5, where the final objective function value (obj), the number of iterations (iter) and the consumed CPU time (measured in seconds) are reported. `lam_max` is denoted by $\lambda_{\max}$ in the table. In the `mat_primal` and LADM columns, *rdif* is defined by

$$rdif = \frac{f_{method} - f_{apgl}}{|f_{apgl}|},$$

where f_{apgl} is the function value obtained by APGL, and f_{method} is that obtained by the corresponding method. Therefore, $rdif < 0$ implies that a smaller function value is obtained.

It can be seen from Table 5 that the proposed LADM (4.7) is very competitive with both APGL and `mat_primal` on solving (6.1) with random data. Specifically, in more than half of the tests LADM obtained smaller objective function values than APGL, and the results are only slightly worse than those obtained by `mat_primal`. As for the results on CPU time, we add the following notes. In the original implementation of `mat_primal`, the standard Matlab subroutine `svd` was used for computing SVD. For a fair comparison, we employed a uniform way of

TABLE 5. Comparison results of APGL, `mat_primal` and LADM (4.7).

Parameters		APGL			mat_primal			LADM		
$(\lambda_{\max}, \mu)$	$p/m/n/r$	obj	iter	CPU	rdif	iter	CPU	rdif	iter	CPU
$(10^0, 10^{-2})$	500/2000/100/20	4.3e3	28	12.1	-1.4e-4	3	6.3	1.5e-12	3	0.9
	1500/2000/100/20	7.4e3	28	28.8	-4.4e-4	3	30.6	-5.6e-8	3	1.8
	2500/2000/100/20	8.6e3	28	44.1	-5.5e-4	3	36.3	7.3e-13	3	2.7
	1000/3000/200/50	2.2e4	28	60.2	-1.5e-4	3	25.8	-1.7e-7	3	4.1
	2000/3000/200/50	3.1e4	28	128	-2.8e-4	3	78.4	-3.4e-8	3	6.6
	3000/3000/200/50	3.8e4	28	158	-3.9e-4	3	101	1.1e-13	3	8.8
$(10^0, 10^0)$	500/2000/100/20	4.3e3	28	11.9	4.1e-14	3	6.5	5.7e-13	5	1.4
	1500/2000/100/20	7.4e3	28	36.7	-2.1e-13	3	37.9	6.6e-10	4	3.4
	2500/2000/100/20	8.6e3	28	44.0	-2.3e-14	3	36.0	6.0e-10	4	3.5
	1000/3000/200/50	2.2e4	28	77.5	1.7e-13	3	31.3	2.9e-11	4	7.2
	2000/3000/200/50	3.1e4	28	128	5.4e-13	3	76.9	2.2e-11	4	8.1
	3000/3000/200/50	3.7e4	28	157	3.7e-13	3	99.7	1.8e-11	4	11.3
$(10^0, 10^2)$	500/2000/100/20	3.3e3	28	12.1	1.2e-15	3	6.7	3.1e-8	25	6.8
	1500/2000/100/20	6.4e3	28	36.8	1.4e-16	3	36.0	6.9e-8	25	18.4
	2500/2000/100/20	7.6e3	31	49.4	8.4e-16	3	35.6	8.5e-8	25	19.6
	1000/3000/200/50	2.0e4	30	61.7	-2.1e-15	3	24.5	1.5e-8	16	20.1
	2000/3000/200/50	2.8e4	29	110	2.3e-15	3	85.1	2.2e-8	16	41.5
	3000/3000/200/50	3.5e4	30	163	-2.5e-15	3	103	2.7e-8	16	41.5
$(10^1, 10^{-2})$	500/2000/100/20	4.3e3	35	24.7	-8.8e-4	60	9.1	2.4e-5	55	15.5
	1500/2000/100/20	7.4e3	36	98.4	-6.2e-4	60	42.9	6.8e-5	66	35.7
	2500/2000/100/20	8.6e3	36	223	-7.4e-4	60	70.1	3.6e-5	68	71.8
	1000/3000/200/50	2.2e4	37	171	-2.5e-4	60	39.2	7.7e-5	56	71.2
	2000/3000/200/50	3.1e4	37	458	-6.4e-4	60	126	5.4e-5	61	135
	3000/3000/200/50	3.8e4	39	887	-6.7e-4	60	190	3.8e-5	67	273
$(10^1, 10^0)$	500/2000/100/20	4.3e3	36	28.4	-3.8e-5	60	8.9	-2.8e-5	54	17.5
	1500/2000/100/20	7.4e3	34	111	-7.9e-5	60	41.4	-7.1e-5	66	41.4
	2500/2000/100/20	8.6e3	34	269	-6.3e-5	60	69.0	-4.1e-5	68	62.5
	1000/3000/200/50	2.2e4	34	171	-7.4e-5	60	40.1	-6.6e-5	56	71.5
	2000/3000/200/50	3.1e4	34	449	-9.0e-5	60	126	-3.5e-5	60	141
	3000/3000/200/50	3.8e4	34	810	-1.1e-4	60	192	-8.1e-5	67	198
$(10^1, 10^2)$	500/2000/100/20	4.0e3	40	26.1	-7.5e-7	50	8.6	-8.4e-7	29	7.6
	1500/2000/100/20	7.1e3	41	123	2.7e-6	30	37.1	-7.4e-6	32	17.1
	2500/2000/100/20	8.3e3	42	225	8.2e-6	30	60.9	-3.1e-6	32	25.7
	1000/3000/200/50	2.1e4	41	190	-6.2e-6	60	39.9	-6.5e-6	38	48.7
	2000/3000/200/50	3.0e4	41	481	3.0e-6	30	110	-7.3e-6	40	81.1
	3000/3000/200/50	3.7e4	40	866	-1.5e-5	30	152	-2.4e-5	41	132
$(10^2, 10^{-2})$	500/2000/100/20	4.4e3	64	35.3	-2.0e-2	300	19.4	-4.4e-3	85	20.8
	1500/2000/100/20	8.8e3	72	119	-1.6e-1	200	67.0	-1.5e-1	116	63.7
	2500/2000/100/20	1.1e4	72	214	-1.9e-1	210	117	-1.7e-1	127	123
	1000/3000/200/50	2.2e4	88	240	-1.2e-2	310	109	-1.1e-2	86	111
	2000/3000/200/50	4.1e4	72	567	-2.5e-1	310	303	-2.3e-1	104	229
	3000/3000/200/50	5.0e4	74	1011	-2.5e-1	300	517	-2.3e-1	120	361
$(10^2, 10^0)$	500/2000/100/20	4.3e3	56	32.5	-3.3e-3	210	15.5	-3.1e-3	85	21.5
	1500/2000/100/20	7.4e3	64	120	-2.6e-3	190	64.8	-2.5e-3	115	58.5
	2500/2000/100/20	8.7e3	63	225	-3.4e-3	190	111	-3.2e-3	127	125
	1000/3000/200/50	2.2e4	59	214	-4.2e-3	200	75.1	-3.6e-3	86	136
	2000/3000/200/50	3.1e4	57	605	-2.9e-3	200	282	-2.5e-3	104	227
	3000/3000/200/50	3.8e4	57	940	-3.1e-3	190	410	-2.8e-3	120	415
$(10^2, 10^2)$	500/2000/100/20	4.2e3	60	31.3	-3.4e-4	190	14.7	-3.4e-4	67	17.0
	1500/2000/100/20	7.3e3	62	137	-5.7e-4	170	62.1	-5.7e-4	89	45.6
	2500/2000/100/20	8.6e3	80	312	-8.9e-5	170	97.3	-8.8e-5	99	75.2
	1000/3000/200/50	2.2e4	58	205	-8.8e-4	180	69.7	-8.8e-4	74	110
	2000/3000/200/50	3.1e4	64	504	-7.5e-4	180	205	-7.4e-4	92	258
	3000/3000/200/50	3.7e4	64	839	-8.6e-4	180	387	-8.5e-4	105	263

computing SVD for all the three compared methods. Specifically, we used the efficient Matlab Mex interface `mexsvd`[2] which computes SVD via a divide-and-conquer routine (`dgesdd`) implemented in LAPACK. Note that this modification can be easily realized in the `mat_primal` code, while for the APGL it can be accomplished by setting `matrix_format` = "standard" and `par.fullsvd` = 1. Since different ways of computing SVD affect CPU time to some extent, the results presented here are only to give a rough experience about how these methods perform. We employed this uniform way of computing SVD just for comparison purpose. In this circumstance, it can be seen from Table 5 that the LADM (4.7) consumed less CPU time than the APGL to obtain comparable objective function values. In comparison with `mat_primal`, the LADM (4.7) is also faster for about half of the tested problems, at the expense of resulting in slightly worse objective function values. As for the results on the number of iterations, it can be seen from Table 5 that all of these three methods in comparison took more iterations as $\lambda_{\max}$ increases. The increasing speed of iteration numbers taken by both APGL and LADM is moderate, while that for `mat_primal` is relatively faster. The performance of all compared methods keeps deteriorating as $\lambda_{\max}$ increases, and `mat_primal` behaves the best for large values of $\lambda_{\max}$ in the sense that it usually obtains the smallest function values. The favorable performance of `mat_primal` is because the accelerated proximal gradient method is applied to a reduced problem of (6.1) which usually has much smaller size. We also note that `mat_primal` needs some nontrivial pre- and post-processing computations such as problem reformulation and solution reconstruction, etc. This is why in some tests `mat_primal` takes longer CPU time than LADM while the iteration number is smaller.

Roughly speaking, (1.2) becomes more difficult as μ decreases. In fact, (1.2) can be viewed as a penalized surrogate of (1.1), and it approaches to (1.1) while $\mu \to 0$. Therefore, small values of μ usually cause typical numerical difficulties which are encountered by penalty type methods. This is the reason why certain continuation and line-search strategies are usually employed to alleviate the difficulty caused by small values of μ (see [20, 49]). We note that the performance of LADM is hardly affected by the value of μ. This can be partially observed from the results in Table 5. Moreover, it is easy to see that by directly setting μ to be 0 the LADM framework (4.7) reduce to (3.7), which indicates that (4.7) converges for $\mu = 0$. From the results in subsections 6.2 and 6.3, the LALM scheme (3.7) converges very well for matrix completion and partial DCT problems.

An important advantage of those algorithms based on the shrinkage operator (2.6) is that the yielded solutions have low-ranks. This is because those singular values smaller than a threshold are shrunk to zero automatically. We note that in all the tests in Table 5, as well as the tests in subsections 6.2 and 6.3 when the sample ratios are relatively high (relative to the rank of X^*), the solutions recovered by LADMs have the same ranks as the original low-rank matrices. This is also true for APGL. However, we also observed that the results recovered by `mat_primal` usually have many small singular values for some test problems, e.g., for the test on $(p, m, n, r, \lambda_{\max}, \mu) = (1000, 2000, 100, 20, 10, 10^{-2})$, the solution recovered by `mat_primal` has rank 100 (even when stopping tolerance is restrictive, e.g., `opt.tol` $= 10^{-8}$ in `mat_primal`), while the true rank of X^* is 20. After applying a post-processing procedure with an appropriate threshold parameter (not

[2]Available in the NNLS package containing the APGL code.

difficult to determine since the singular values are usually clustered into two groups, where the magnitudes of one group are frequently significantly bigger than those of the other group), the "correct" rank of X^* can then be identified. Finally, we note that, for all the tests presented in Table 5, the resulting values of $\|AX-B\|_F/\|B\|_F$ are in the order of $O(10^{-3})$ or smaller, and are compatible to our synthetic noise level.

6.5. **Summary.** From the numerical results in subsections 6.2 and 6.3, the proposed LALM and LADMs perform promisingly, and they are even competitive to the ADMs for the special case where $\mathcal{A}\mathcal{A}^* = I$. The favorable performance of LALM and LADMs on these random problems are mainly because the measurement system is well conditioned. In fact, $\mathcal{A}\mathcal{A}^* = \mathcal{I}$ holds for matrix completion and partial DCT problems. For the general case with a generic linear operator that $\mathcal{A}\mathcal{A}^* \neq I$, the results in Section 6.4 also show that the proposed LADM scheme (4.7) is competitive to existing methods in the literature. Specifically, the LADM (4.7) demonstrates stable and effective convergence for random problems with different conditions. As the condition number $\lambda_{\max}(A^\top A)$ (for problem (6.1)) increases, the same as for the compared methods, the LADM (4.7) requires more iterations to achieve a solution of certain accuracy. Although the proposed LALM and LADMs are not necessarily always the fastest (again, for matrix completion problems with small sample ratios relative to $rank(X^*)$, the APGL performs better, see [10]; and for the special case (6.1), the algorithms in [38] are faster in some cases due to their ability of reducing the problem dimensionality), our experimental results convincingly demonstrate that they converge very well for all three problems (1.1)-(1.3) under consideration, and they are very competitive to the APGL and `mat_primal`, both of which are only customized for solving a certain model.

6.6. **A note on β.** We note that the penalty parameter β plays a critical rule for the efficiency of the proposed LALM and LADMs (as well as other augmented Lagrangian related methods). Theoretically, a larger value of β leads to faster convergence of the outer loop of the ALM scheme (3.2), see [42]. However, a very large value of β usually causes numerical difficulty, and it is thus not recommended in practice. In general, determining suitable values of β is problem-dependant. Roughly speaking, for solving constrained problems a suitable value of β should be chosen such that, when the constraints are penalized to the objective functions, their magnitudes should be balanced well. Therefore, when implementing these first-order methods in practice, appropriate scaling of the problem data is usually necessary. As such, some empirical way of choosing β can be incorporated, which usually performs well in practice. For the problems studied in this paper, we set $\beta = \beta_0 := 2.5/\min(m,n)$ uniformly in all of the tests. To illustrate the performance of LADM (4.7) with respect to different β values, we plot in Figure 1 the variation of the objective function values of (6.1) with respect to the iteration numbers.

It can be seen from the plot on the left-hand side of Figure 1 ($\lambda_{\max} = 10^2$) that $\beta = \beta_0$ performs the best. For the plot on the right-hand side of Figure 1 ($\lambda_{\max} = 10^3$), the performance of $\beta = \beta_0$ is also among the best ones, although it is slightly slower than the setting $\beta = 5\beta_0$ asymptotically. Surely, the results in Figure 1 are merely illustrative examples, and for other problems this value may not always perform the best. However, this choice of β leads to relatively stable and effective convergence of the proposed LALM and LADMs for all the tested

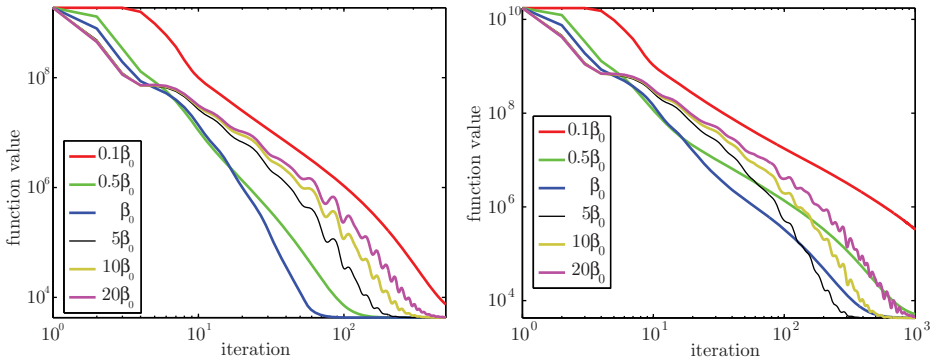

FIGURE 1. Test results on different choices of β. The values of p, m, r and μ for problem (6.1) are, respectively, 500, 2000, 100, 20 and 0.01. Each run is terminated by maximum iteration number `maxit`. Left: $\lambda_{\max} = 10^2$ and `maxit` $= 500$; Right: $\lambda_{\max} = 10^3$ and `maxit` $= 1000$.

problems. In general, for linearly constrained problems the penalty parameter β can be adjusted adaptively, subject to the principle of balancing the residuals of the primal and the dual problems. However, this approach is not always beneficial since the computation of the dual residual could be nontrivial, e.g., the dual problem of (4.1) has a ball constraint determined by the matrix operator norm (the largest singular value). This is why we adopted a simple empirical strategy on selecting β in our experiments. We refer to, e.g., [23, 24, 27], for some self-adaptive rules of tuning β.

7. Conclusions

In this paper, we consider the augmented Lagrangian method (ALM) and the alternating direction method (ADM) for nuclear norm related problems. We first show that the ADM is applicable to all the three models (1.1)-(1.3) provided that the involved linear operator $\mathcal{A}$ satisfies $\mathcal{A}\mathcal{A}^* = \mathcal{I}$. For the general case where $\mathcal{A}\mathcal{A}^* \neq \mathcal{I}$, we derive linearized ALMs and ADMs in a unified manner, i.e., linearizing the difficult subproblems such that closed-form solutions of the linearized problems can be derived. Global convergence of these methods are established under standard assumptions. In particular, we establish the global convergence for the linearized ALM (3.7), which is stronger than those in [58]. An advantage of the proposed LALM and LADMs is that they are easily implementable to all the models (1.1)-(1.3), where $\mathcal{A}$ can be a generic linear operator, while previous approaches are mostly customized for a particular model. Extensive numerical results on various data show that the proposed linearized methods perform promisingly and they are even very competitive to the customized algorithms APGL for (1.2) and `mat_primal` for (6.1).

Finally, we note that the algorithms derived in this paper can be easily extended to general problems with $J(X)$ in place of $\|X\|_*$, where $J(X)$ can be a generic closed proper convex function. The resulting algorithms are easily implementable provided that the solution to a problem of the form (2.7) can be easily computed. Clearly, this will include, but not limited to, the cases of the ℓ_1-, ℓ_2- and ℓ_∞-norms

in vector spaces, which have many applications in, e.g., compressive sensing and machine learning, etc.

Acknowledgement

We are grateful to two anonymous referees for their valuable comments and suggestions which have helped us improve the presentation of this paper substantially.

References

1. J. Abernethy, F. Bach, T. Evgeniou and J.-P. Vert, *A new approach to collaborative filtering: Operator estimation with spectral regularization*, Journal of Machine Learning Research, 10(2009), pp. 803–826.
2. A. Argyriou, T. Evgeniou and M. Pontil, *Convex multi-task feature learning*, Machine Learning, 73(3)(2008), pp. 243–272.
3. J. M. Borwein, and A. S. Lewis, *Convex analysis and nonlinear optimization*, Springer-Verlag, 2003. MR2184742 (2006f:49001)
4. J. F. Cai, E. J. Candés and Z. W. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM Journal on Optimization, 20(4) (2010), pp. 1956–1982. MR2600248 (2011c:90065)
5. J. F. Cai, S. Osher and Z. Shen, *Linearized Bregman iterations for compressed sensing*, Mathematics of Computation, 78(267)(2009), pp. 1515–1536. MR2501061 (2010e:65086)
6. J. F. Cai, S. Osher and Z. Shen, *Convergence of the Linearized Bregman Iteration for ℓ_1-Norm Minimization*, Mathematics of Computation, 78(268) (2009), pp. 2127–2136. MR2521281 (2010k:65111)
7. J. F. Cai, S. Osher and Z. Shen, *Split Bregman Methods and Frame Based Image Restoration*, Multiscale Model. Simul., 8(2)(2009), pp. 337–369. MR2581025 (2011a:94016)
8. E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, 9(2009), pp. 717–772. MR2565240 (2011c:90066)
9. E. J. Candès and T. Tao, *The power of convex relaxation: near-optimial matrix completion*, IEEE Transactions on Information Theory, 56(5) (2009), pp. 2053-2080. MR2723472
10. C. H. Chen, B. S. He and X. M. Yuan, *Matrix completion via alternating direction methods*, IMA Journal of Numerical Analysis, to appear.
11. P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200. MR2203849 (2007g:94016)
12. J. Eckstein and M. Fukushima, *Some reformulation and applications of the alternating directions method of multipliers*, In: Hager, W. W. *et al.* eds., Large Scale Optimization: State of the Art, Kluwer Academic Publishers, pp. 115–134, 1994. MR1307168
13. E. Esser, *Applications of Lagrangian-based alternating direction methods and connections to split Bregman*, preprint, available at `http://www.math.ucla.edu/applied/cam/`, 2009.
14. M. Fazel, H. Hindi and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, Proceedings American Control Conference, 6(2001), pp. 4734–4739.
15. M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming problems*, Computational Optimization and Applications, 1(1992), pp. 93–111. MR1195631 (94a:90020)
16. D. Gabay, *Application of the method of multipliers to varuational inequalities*, In: Fortin, M., Glowinski, R., eds., Augmented Lagrangian methods: Application to the numerical solution of Boundary-Value Problem, North-Holland, Amsterdam, The Netherlands, pp. 299–331, 1983.
17. D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximations*, Computational Mathematics with Applications, 2(1976), pp. 17–40.
18. R. Glowinski and P. Le Tallec, Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics, SIAM Studies in Applied Mathematics, Philadelphia, PA, 1989. MR1060954 (91f:73038)
19. T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM Journal on Imaging Science, 2(2) (2009), pp. 323–343. MR2496060 (2010e:65087)

20. E. Hale, W. Yin and Y. Zhang, *Fixed-point continuation for ℓ_1-minimization: methodology and convergence*, SIAM Journal on Optimization, 19(3) (2008), pp.1107–1130. MR2460734 (2009j:90070)
21. B. S. He, L. Z. Liao, D. Han and H. Yang, *A new inexact alternating directions method for monontone variational inequalities*, Mathematical Programming, 92(2002), pp. 103–118. MR1892298 (2003b:90111)
22. B. S. He, M. H. Xu and X. M. Yuan, *Solving large-scale least squares semidefinite programming by alternating direction methods*, SIAM Journal on Matrix Analysis and Applications, 32(1) (2011), pp. 136–152. MR2811295
23. B. S. He and H. Yang, *Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities*, Operations Research Letters, 23 (1998), pp. 151–161. MR1677664 (2000d:90089)
24. B. S. He, H. Yang and S. L. Wang, *Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities*, Journal of Optimization theory and applications, 106 (2000), pp. 337–356. MR1788928 (2001h:49016)
25. M. R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, 4 (1969), pp. 303–320. MR0271809 (42:6690)
26. S. Ji and J. Ye, *An accelerated gradient method for trace norm minimization*, The Twenty-Sixth International Conference on Machine Learning, 2009.
27. S. Kontogiorgis and R. R. Meyer, *A variable-penalty alternating directions method for convex optimization*, Mathematical Programming, 83(1998), pp. 29–53. MR1643963 (99k:90116)
28. R. M. Larsen, *PROPACK-Software for large and sparse SVD calculations*, Available at: `http://sun.stanfor.edu/srmunk/PROPACK/`, 2005.
29. Z.-C. Lin, M.-M. Chen, L.-Q. Wu and Y. Ma, *The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices*, manuscript, 2009.
30. J. Liu, S. Ji and J. Ye, *SLEP: A Sparse Learning Package*, Version 2.0, Available at: `http://www.public.asu.edu/~jye02/Software/SLEP`, 2010.
31. Y. J. Liu, D. F. Sun and K. C. Toh, *An implementable proximal point algorithmic framework for nuclear norm minimization*, Mathematical Programming, to appear.
32. S. Q. Ma, D. Goldfarb and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., 128 (2011), 321–353. MR2810961
33. Y. E. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Doklady AN SSSR, 269, pp. 543–547, 1983. MR701288 (84i:90119)
34. Y. E. Nesterov, *Smooth minimization of nonsmooth functions*, Mathematical Programming, 103 (2005), pp. 127–152. MR2166537 (2006g:90174)
35. M. K. Ng, P. A. Weiss and X. M. Yuan, *Solving constrained total-variation problems via alternating direction methods*, SIAM Journal on Scientifc Computing, 32(5) (2010), pp. 2710–2736. MR2684734 (2011i:65065)
36. G. Obozinski, B. Taskar and M. I. Jordan, *Joint covariate selection and joint subspace selection for multiple classification problems*, Statistics and Computing, 2009. MR2610775
37. S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul. 4(2) (2005), pp. 460–489. MR2162864 (2006c:49051)
38. T. K. Pong, P. Tseng, S. Ji. and J. Ye, *Trace norm regularization: reformulations, algorithms, and multi-task learning*, SIAM Journal on Optimization, 20 (2010), pp. 3465-3489. MR2763512
39. M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, NY, pp. 283–298, 1969. MR0272403 (42:7284)
40. B. Recht, M. Fazel and P. A. Parrilo, *Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization*, SIAM Review, 52(2010), pp. 471–501. MR2680543
41. R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, NJ, 1970. MR0274683 (43:445)
42. R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of Operations Research, 1 (1976), 97–116. MR0418919 (54:6954)

43. S. Setzer, G. Steidl and T. Teuber, *Deblurring Poissonian images by split Bregman techniques*, Journal of Visual Communication and Image Representation, 21 (2010), pp. 193-199.
44. N. Srebro, J. D. M. Rennie and T. S. Jaakkola, *Maximum-margin matrix factorization*, Advances in Neural Information Processing System, (2005), pp. 1329–1336.
45. J. F. Sturm, *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software 11 & 12 (1999), pp. 625–653. MR1778433
46. W. Y. Sun and Y. X. Yuan, *Optimization theory and methods*, Nonlinear Programming Series: Springer Optimization and Its Applications, 2006. MR2232297 (2007c:90002)
47. J. Sun and S. Zhang, *A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs*, European Journal of Operational Research, 207 (2010), pp. 1210-1220. MR2727074
48. 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) (2011), pp. 57–81. MR2765489
49. K. C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least suqares problems*, Pacific Journal of Optimization, 6(2010), pp. 615–640. MR2743047
50. P. Tseng, *Alternating projection-proximal methods for convex programming and variational inequalities*, SIAM Journal on Optimization, 7(1997), pp. 951–965. MR1479608 (99a:90156)
51. R. H. Tütüncü, K. C. Toh and M. J. Todd, *Solving semidefinite-quadrtic-linear programs using SDPT3*, Mathematical Programming, 95(2003), pp. 189–217. MR1976479 (2004c:90036)
52. G. A. Watson, *Characterization of the subdifferential of some matrix norms*, Linear Algebra and its Applications, 170 (1992), pp. 33–45. MR1160950 (93b:15031)
53. Z. W. Wen, D. Goldfarb and W. Yin, *Alternating direction augmented Lagrangina methods for semidefinite programming*, Mathematical Programming Computation, 2 (2010), pp. 203–230. MR2741485
54. Z. W. Wen, W. Yin and Y. Zhang, *Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm*, TR10-07, CAAM, Rice University, 2010.
55. J.-F. Yang and Y. Zhang, *Alternating direction method for ℓ_1-problems in compressive sensing*, SIAM Journal on Scientific Computing, 33(1) (2011), pp. 250–278. MR2783194
56. W. Yin, *Analysis and generalizations of the linearized Bregman method*, SIAM Journal on Imaging Sciences, 3(4) (2010), pp. 856–877. MR2735964 (2011j:68172)
57. W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing.* SIAM Journal on Imaging Science, 1 (2008), pp. 143–168. MR2475828 (2010f:90170)
58. X. Zhang, M. Burger, X. Bresson and S. Osher, *Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction*, SIAM Journal on Imaging Science, 3(2010), pp. 253–276. MR2679428

Department of Mathematics, Nanjing University, 22 Hankou Road, Nanjing, 210093, People's Republic of China.

E-mail address: jfyang@nju.edu.cn

Department of Mathematics, Hong Kong Baptist University, Hong Kong, People's Republic of China.

E-mail address: xmyuan@hkbu.edu.hk