SIAM J. OPTIM.
Vol. 22, No. 2, pp. 313–340

ALTERNATING DIRECTION METHOD WITH GAUSSIAN BACK SUBSTITUTION FOR SEPARABLE CONVEX PROGRAMMING*

BINGSHENG HE†, MIN TAO†, AND XIAOMING YUAN‡

Abstract. We consider the linearly constrained separable convex minimization problem whose objective function is separable into m individual convex functions with nonoverlapping variables. A Douglas–Rachford alternating direction method of multipliers (ADM) has been well studied in the literature for the special case of $m = 2$. But the convergence of extending ADM to the general case of $m \geq 3$ is still open. In this paper, we show that the straightforward extension of ADM is valid for the general case of $m \geq 3$ if it is combined with a Gaussian back substitution procedure. The resulting ADM with Gaussian back substitution is a novel approach towards the extension of ADM from $m = 2$ to $m \geq 3$, and its algorithmic framework is new in the literature. For the ADM with Gaussian back substitution, we prove its convergence via the analytic framework of contractive-type methods, and we show its numerical efficiency by some application problems.

Key words. alternating direction method, convex programming, Gaussian back substitution, separable structure

AMS subject classifications. 90C25, 65K05, 94A08

DOI. 10.1137/110822347

1. Introduction.

In the literature, the Douglas–Rachford alternating direction method of multipliers (ADM) proposed originally in [15] (see also [14]) has been well studied for the following linearly constrained separable convex minimization problem, whose objective function is separated into two individual convex functions with nonoverlapping variables:

$$
\begin{aligned}
\min \quad & \theta_1(x_1) + \theta_2(x_2), \\
& A_1 x_1 + A_2 x_2 = b, \\
& x_1 \in \mathcal{X}_1 \quad \text{and} \quad x_2 \in \mathcal{X}_2,
\end{aligned}
\tag{1.1}
$$

where $\theta_1 : \Re^{n_1} \to \Re$ and $\theta_2 : \Re^{n_2} \to \Re$ are closed proper convex functions (not necessarily smooth), $\mathcal{X}_1 \subseteq \Re^{n_1}$ and $\mathcal{X}_2 \subseteq \Re^{n_2}$ are closed convex sets, $A_1 \in \Re^{l \times n_1}$ and $A_2 \in \Re^{l \times n_2}$ are given matrices, and $b \in \Re^l$ is a given vector. We refer the reader to, e.g., [11, 13, 16, 20, 28, 45], for some early references on ADM. More specifically, the iterative scheme of ADM for solving (1.1) is

$$
\left\{
\begin{aligned}
x_1^{k+1} &= \arg\min \left\{ \theta_1(x_1) + \tfrac{\beta}{2} \|(A_1 x_1 + A_2 x_2^k - b) - \tfrac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_1 \in \mathcal{X}_1 \right\}, \\
x_2^{k+1} &= \arg\min \left\{ \theta_2(x_2) + \tfrac{\beta}{2} \|(A_1 x_1^{k+1} + A_2 x_2 - b) - \tfrac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_2 \in \mathcal{X}_2 \right\}, \\
\lambda^{k+1} &= \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b),
\end{aligned}
\right.
\tag{1.2}
$$

*Received by the editors January 26, 2011; accepted for publication (in revised form) January 23, 2012; published electronically April 12, 2012.
http://www.siam.org/journals/siopt/22-2/82234.html

†Department of Mathematics, Nanjing University, Nanjing, Jiangsu, 210093, People's Republic of China (hebma@nju.edu.cn, taomin0903@gmail.com). The first author was supported by NSFC grants 10971095 and 91130007, the Cultivation Fund of KSTIP-MOEC 708044, and the MOEC fund 20110091110004.

‡Corresponding author. Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong, People's Republic of China (xmyuan@hkbu.edu.hk). This author was supported by the General Research Fund of Hong Kong: HKBU203311.

where $\lambda^k \in \Re^l$ is the Lagrange multiplier associated with the linear constraint and $\beta > 0$ is the penalty parameter for the violation of the linear constraint.

If (1.1) is treated as a generic linearly constrained convex minimization problem and its separable structure is ignored, the classical augmented Lagrangian method (ALM) [26, 34] can be applied directly. The ADM scheme (1.2), however, decomposes the subproblem of ALM into two subproblems in Gauss–Seidel fashion at each iteration, and thus the variables x_1 and x_2 can be solved separately in alternating order. For many concrete applications of (1.1), the individual functions $\theta_1(x_1)$ and $\theta_2(x_2)$ both have specific properties, and the decomposition treatment of ADM makes it possible to exploit these particular properties separately. The decomposed subproblems in (1.2) are thus often simple enough to have closed-form solutions or can be easily solved up to high precision. We refer the reader to, e.g., [8, 12, 25, 32, 38, 39, 40, 43, 44] and references cited therein, for some novel applications of ADM in such diverse areas as image processing, statistical learning, and compressive sensing.

In this paper, we consider a general case of the linearly constrained separable convex minimization problem with $m \geq 3$,

$$\begin{array}{ll} \min & \sum_{i=1}^{m} \theta_i(x_i), \\ & \sum_{i=1}^{m} A_i x_i = b, \\ & x_i \in \mathcal{X}_i, \quad i = 1, \ldots, m, \end{array} \tag{1.3}$$

where $\theta_i : \Re^{n_i} \to \Re$ $(i = 1, \ldots, m)$ are closed proper convex functions (not necessarily smooth), $\mathcal{X}_i \subseteq \Re^{n_i}$ $(i = 1, \ldots, m)$ are closed convex sets, $A_i \in \Re^{l \times n_i}$ $(i = 1, \ldots, m)$ are given matrices, and $b \in \Re^l$ is a given vector. Throughout, we assume that the matrices $A_i^T A_i$ $(i = 1, \ldots, m)$ are nonsingular and the solution set of (1.3) is nonempty. Note that although we restrict our analysis to the case of (1.3) with vector variables, all of the following results can be straightforwardly extended to the case with matrix variables (see section 5.1).

Our consideration of the extension from (1.1) to (1.3) is motivated by a number of concrete applications; see, e.g., [4, 3, 41, 31, 38]. Inspired by the efficiency of ADM, a natural idea for solving (1.3) is to extend the ADM scheme (1.2) from the special case (1.1) to the general case (1.3), and this straightforward extension results in the following ADM iterative scheme.

$$\left\{ \begin{array}{l} x_1^{k+1} = \arg\min \left\{ \theta_1(x_1) + \frac{\beta}{2} \|(A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b) - \frac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_1 \in \mathcal{X}_1 \right\}, \\ x_2^{k+1} = \arg\min \Big\{ \theta_2(x_2) + \frac{\beta}{2} \|(A_1 x_1^{k+1} + A_2 x_2 + \sum_{j=3}^m A_j x_j^k - b) \\ \qquad\qquad\qquad - \frac{1}{\beta}\lambda^k\|^2 \big| x_2 \in \mathcal{X}_2 \Big\}, \\ \quad \vdots \\ x_i^{k+1} = \arg\min \Big\{ \theta_i(x_i) + \frac{\beta}{2} \|(\sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i + \sum_{j=i+1}^m A_j x_j^k - b) \\ \qquad\qquad\qquad - \frac{1}{\beta}\lambda^k\|^2 \big| x_i \in \mathcal{X}_i \Big\}, \\ \quad \vdots \\ x_m^{k+1} = \arg\min \Big\{ \theta_m(x_m) + \frac{\beta}{2} \|(\sum_{j=1}^{m-1} A_j x_j^{k+1} + A_m x_m - b) \\ \qquad\qquad\qquad - \frac{1}{\beta}\lambda^k\|^2 \;\big|\; x_m \in \mathcal{X}_m \Big\}, \\ \lambda^{k+1} = \lambda^k - \beta(\sum_{j=1}^m A_j x_j^{k+1} - b). \end{array} \right. \tag{1.4}$$

Since the scheme (1.4) is obtained via splitting the $(k+1)$th ALM subproblem of (1.3) alternatingly, we follow [40] and name (1.4) as the alternating splitting augmented Lagrangian method (ASALM). Obviously, the ASALM scheme (1.4) can exploit each particular property of $\theta_i(x_i)$ individually, and thus the advantage of the original ADM (1.2) is preserved fully for solving (1.3). In fact, all of the x_i-related subproblems in (1.4) are in the form of

$$\min\left\{\theta_i(x_i) + \frac{\beta}{2}\|A_i x_i - a_i\|^2 \mid x_i \in \mathcal{X}_i\right\} \tag{1.5}$$

with certain known $a_i \in \Re^l$. Note that for many applications the subproblems (1.5) are often easy because of the simplicity of A_i's and $\theta_i(x_i)$'s, e.g., the least-squares data-fitting function, the l_1 regularization [9], the l_2 Tikhonov regularization [42], and the nuclear norm to induce low-rank solutions [6, 36]. Thus, the desire to extend ADM from (1.2) to (1.4) is rational.

The convergence of ASALM, however, has not yet been proved theoretically, even though its efficiency has been verified empirically by some recent applications (see [33, 40]). In fact, even for the special case of (1.3) with $m = 3$, the convergence of ASALM is still open. We refer the reader to [18, 21, 22] for some recent efforts in extending ADM to the general case (1.3).

In this paper, we provide a novel approach towards the extension of ADM for (1.3). More specifically, we show that if a new iterate is generated by correcting the output of (1.4) with a Gaussian back substitution procedure, then the resulting sequence of such iterates converges to a solution of (1.3). From now on, the resulting method is called the ADM with Gaussian back substitution (ADM-G). In essence, ADM-G predicts the new iterate in the forward order ($x_1^{k+1} \to x_2^{k+1} \to \cdots \to x_m^{k+1} \to \lambda^{k+1}$) via the ASALM (1.4), and then corrects the predictor in the backward order ($\lambda^{k+1} \to x_m^{k+1} \to x_{m-1}^{k+1} \to \cdots \to x_2^{k+1} \to x_1^{k+1}$) via a Gaussian back substitution procedure. In this sense, each iteration of ADM-G consists of a forward procedure (ADM procedure) and a backward procedure (Gaussian back substitution procedure). Alternatively, ADM-G can be regarded as a prediction-correction type of method, where the predictor is generated by ASALM and the correction is completed by a Gaussian back substitution procedure. We prove the convergence of ADM-G under the analytic framework of contractive-type methods [2] and show its numerical efficiency via several concrete applications of (1.3) in various disciplines.

The rest of the paper is organized as follows. In section 2, we characterize (1.3) by a variational reformulation, which is convenient for further analysis. Then, in section 3, we present the iterative scheme of ADM-G and make some remarks. Implementation details of the Gaussian back substitution procedure for some special cases of (1.3) will also be delineated. In section 4, we prove the global convergence of ADM-G. In section 5, we apply ADM-G to solve some concrete applications of (1.3) and compare it with some existing methods numerically. Finally, we offer some conclusions in section 6.

2. Variational inequality characterization of (1.3).

In this section, we derive a variational reformulation of (1.3) which will be used in future analysis.

By attaching a Lagrange multiplier vector $\lambda \in \Re^l$ to the linear constraint, the Lagrange function of (1.3) is given by

$$L(x_1, x_2, \ldots, x_m, \lambda) = \sum_{i=1}^{m} \theta_i(x_i) - \lambda^T \left(\sum_{i=1}^{m} A_i x_i - b \right), \tag{2.1}$$

and it is defined on the set

$$\mathcal{W} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l.$$

Let $(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*)$ be a saddle point of the Lagrange function (2.1). Then, for any $\lambda \in \Re^l$ and $x_i \in \mathcal{X}_i$ $(i = 1, 2, \ldots, m)$, we have

$$L(x_1^*, x_2^*, \ldots, x_m^*, \lambda) \le L(x_1^*, x_2^*, \ldots, x_m^*, \lambda^*) \le L(x_1, x_2, \ldots, x_m, \lambda^*).$$

For $i \in \{1, 2, \ldots, m\}$, we denote by $\partial\theta_i(x_i)$ the subdifferential of the convex function $\theta_i(x_i)$, and we use $f_i(x_i) \in \partial\theta_i(x_i)$ to denote a subgradient of $\theta_i(x_i)$. Then, finding a saddle point of $L(x_1, x_2, \ldots, x_m, \lambda)$ is equivalent to finding $w^* = (x_1^*, x_2^*, \ldots, x_m^*, \lambda^*) \in \mathcal{W}$ and $f_i(x_i^*) \in \partial\theta_i(x_i^*)$ (for $i = 1, 2, \ldots, m$) such that the following inequalities are satisfied:

$$\left\{ \begin{array}{l} (x_1 - x_1^*)^T \{f_1(x_1^*) - A_1^T \lambda^*\} \ge 0, \\ (x_2 - x_2^*)^T \{f_2(x_2^*) - A_2^T \lambda^*\} \ge 0, \\ \qquad \vdots \\ (x_m - x_m^*)^T \{f_m(x_m^*) - A_m^T \lambda^*\} \ge 0, \\ (\lambda - \lambda^*)^T (\sum_{i=1}^m A_i x_i^* - b) \ge 0, \end{array} \right. \qquad \forall\, w = (x_1, x_2, \ldots, x_m, \lambda) \in \mathcal{W}. \tag{2.2}$$

We denote by $\mathcal{W}^*$ the set of such w^* that satisfies (2.2). Then, under the aforementioned nonempty assumption on the solution set of (1.3), obviously $\mathcal{W}^*$ is also nonempty.

3. ADM with Gaussian back substitution.

In this section, we show the combination of the ASALM (1.4) with a Gaussian back substitution procedure and derive the resulting ADM-G for solving (1.3). We also elucidate how to realize the Gaussian back substitution for some special cases of (1.3).

To present the Gaussian back substitution procedure, we first define some matrices which will be frequently used for later analysis. More specifically, let

$$M = \begin{pmatrix} \beta A_2^T A_2 & 0 & \cdots & \cdots & 0 \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix}, \tag{3.1}$$

$$Q = \begin{pmatrix} \beta A_2^T A_2 & \beta A_2^T A_3 & \cdots & \beta A_2^T A_m & A_2^T \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \cdots & \beta A_3^T A_m & A_3^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & A_m^T \\ A_2 & A_3 & \cdots & A_m & \frac{1}{\beta} I_l \end{pmatrix}, \tag{3.2}$$

and

$$H = \operatorname{diag}\left(\beta A_2^T A_2, \beta A_3^T A_3, \ldots, \beta A_m^T A_m, \frac{1}{\beta} I_l\right). \tag{3.3}$$

Note that for any $\beta > 0$, under the assumption that all the $A_i^T A_i$ matrices are nonsingular, the matrix M defined in (3.1) is a nonsingular lower-triangular block matrix, and the matrix H defined in (3.3) is a symmetric positive definite matrix. In addition, according to (3.1) and (3.3), we easily have

$$H^{-1}M^T = \begin{pmatrix} I_{n_2} & (A_2^T A_2)^{-1} A_2^T A_3 & \cdots & (A_2^T A_2)^{-1} A_2^T A_m & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & (A_{m-1}^T A_{m-1})^{-1} A_{m-1}^T A_m & 0 \\ 0 & \cdots & 0 & I_{n_m} & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix}, \tag{3.4}$$

which is an upper-triangular block matrix, and its diagonal components are all identity matrices. We would highlight that the Gaussian back substitution procedure to be proposed is based on the matrix $H^{-1}M^T$ defined in (3.4).

Before presenting the iterative scheme of ADM-G, we first clarify some notation for the convenience of further analysis. With the given w^k, we will use the notation $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{x}_3^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ to denote the predictor which is generated by the ASALM scheme (1.4), and the new iterate (i.e., the corrector) will be denoted by $w^{k+1} = (x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \ldots, x_m^{k+1}, \lambda^{k+1})$. Moreover, by revisiting the iterative scheme of the original ADM (1.2) and the ASALM (1.4), it is easy to observe that the variable x_1 plays only an intermediate role and is not involved in the execution of (1.2) or (1.4); e.g., see the elaboration in [4]. Therefore, the input for executing the iteration of (1.4) is only the sequence $\{x_2^k, x_3^k, \ldots, x_m^k, \lambda^k\}$. For this reason, we define the following notation, which will simplify our analysis:

$$\begin{gathered} v = (x_2, \ldots, x_m, \lambda), \quad \mathcal{V} = \mathcal{X}_2 \times \cdots \times \mathcal{X}_m \times \Re^l, \\ v^k = (x_2^k, \ldots, x_m^k, \lambda^k), \quad \tilde{v}^k = (\tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k) \ \forall\ k \in \mathcal{N}, \\ v^* = (x_2^*, \ldots, x_m^*, \lambda^*), \quad \mathcal{V}^* = \{(x_2^*, \ldots, x_m^*) \,|\, (x_1^*, x_2^*, \ldots, x_m^*, \lambda^*) \in \mathcal{W}^*\}. \end{gathered}$$

Now we are ready to propose the iterative scheme of ADM-G for solving (1.3).

Algorithm (the ADM (ADM-G) (1.3)). Let $\beta > 0$ and $\alpha \in (0, 1)$, and let the matrices M and H be defined by (3.1) and (3.3), respectively. With the given iterate w^k, the new iterate w^{k+1} is generated as follows.

Step 1. ADM step (prediction step). Obtain $\tilde{w}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \ldots, \tilde{x}_m^k, \tilde{\lambda}^k)$ in the forward (alternating) order by executing the ASALM (1.4):

(3.5a)

$$\begin{cases} \tilde{x}_1^k = \arg\min\left\{\theta_1(x_1) + \frac{\beta}{2}\|(A_1x_1 + \sum_{j=2}^m A_jx_j^k - b) - \frac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_1 \in \mathcal{X}_1\right\}, \\ \tilde{x}_2^k = \arg\min\left\{\theta_2(x_2) + \frac{\beta}{2}\|(A_1\tilde{x}_1^k + A_2x_2 + \sum_{j=3}^m A_jx_j^k - b) \right. \\ \qquad\qquad \left. -\frac{1}{\beta}\lambda^k\|^2 \middle| x_2 \in \mathcal{X}_2\right\}, \\ \quad\vdots \\ \tilde{x}_i^k = \arg\min\left\{\theta_i(x_i) + \frac{\beta}{2}\|(\sum_{j=1}^{i-1} A_j\tilde{x}_j^k + A_ix_i + \sum_{j=i+1}^m A_jx_j^k - b) \right. \\ \qquad\qquad \left. -\frac{1}{\beta}\lambda^k\|^2 \middle| x_i \in \mathcal{X}_i\right\}, \\ \quad\vdots \\ \tilde{x}_m^k = \arg\min\left\{\theta_m(x_m) + \frac{\beta}{2}\|(\sum_{j=1}^{m-1} A_j\tilde{x}_j^k + A_mx_m - b) \right. \\ \qquad\qquad \left. -\frac{1}{\beta}\lambda^k\|^2 \;\middle|\; x_m \in \mathcal{X}_m\right\}, \\ \tilde{\lambda}^k = \lambda^k - \beta(\sum_{j=1}^m A_j\tilde{x}_j^k - b). \end{cases}$$

Step 2. Gaussian back substitution step (correction step). Generate the new iterate w^{k+1} by correcting $\tilde{w}^k$ in the backward order:

$$\text{(3.5b)} \qquad \begin{cases} H^{-1}M^T(v^{k+1} - v^k) = \alpha(\tilde{v}^k - v^k), \\ x_1^{k+1} = \tilde{x}_1^k. \end{cases}$$

Remark 1. Recall that the matrix $H^{-1}M^T$ defined in (3.4) is an upper-triangular block matrix. The Gaussian back substitution step (3.5b) is thus easy to execute. In fact, as we have mentioned, after the predictor is generated by the ADM scheme (3.5a) in the forward (alternating) order, the proposed Gaussian back substitution step corrects the predictor in the backward order.

Remark 2. To show the main idea with clearer notation, we restrict our theoretical discussion to the case where $\beta > 0$ is fixed. Some strategies developed in [23, 24] for adjusting the values of β dynamically during iterations can be easily combined with the proposed algorithm.

Remark 3. The reason for the restriction $\alpha \in (0, 1)$ becomes clear in Theorem 4.4 (see (4.21)). In practice, according to our experience, we recommend $\alpha \in [0.5, 1)$ (or even more aggressively, $\alpha = 1$), which may yield faster convergence empirically.

Remark 4. The Gaussian back substitution step (3.5b) can be rewritten as

$$\text{(3.6)} \qquad v^{k+1} = v^k - \alpha M^{-T}H(v^k - \tilde{v}^k).$$

As we will show, for any $v^* \in \mathcal{V}^*$, $-M^{-T}H(v^k - \tilde{v}^k)$ is a descent direction of the distance function $\frac{1}{2}\|v - v^*\|_G^2$ with $G = MH^{-1}M^T$ at the point $v = v^k$. In this sense, the proposed ADM-G can also be regarded as an ADM-based contraction method, where the output of the ADM step (3.5a) contributes a descent direction of the distance function $\frac{1}{2}\|v - v^*\|_G^2$. Thus, the constant α in (3.5b) plays the role of a step size along the descent direction $-M^{-T}H(v^k - \tilde{v}^k)$. In fact, we can choose the step size dynamically based on some techniques in the literature (see, e.g., [45]), and the Gaussian back substitution procedure with the constant α can be modified accordingly into the following form with a dynamical step size:

$$H^{-1}M^T(v^{k+1} - v^k) = \gamma\alpha_k^*(\tilde{v}^k - v^k), \tag{3.7}$$

where $\gamma \in (0, 2)$, Q is defined as in (3.2), and

$$\alpha_k^* := \frac{\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2}{2\|v^k - \tilde{v}^k\|_H^2}. \tag{3.8}$$

Indeed, for any $\beta > 0$, the symmetric matrix Q is positive semidefinite. Then, for a given v^k, let $\tilde{v}^k$ be obtained by the ADM procedure (3.5a), and we then have

$$\|v^k - \tilde{v}^k\|_H^2 = \beta \sum_{i=2}^m \big\|A_i(x_i^k - \tilde{x}_i^k)\big\|^2 + \frac{1}{\beta}\big\|\lambda^k - \tilde{\lambda}^k\big\|^2$$

and

$$\|v^k - \tilde{v}^k\|_Q^2 = \beta \left\| \sum_{i=2}^m A_i(x_i^k - \tilde{x}_i^k) + \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \right\|^2,$$

where the notation $\|x\|_H$ and $\|x\|_Q$ are defined by $(x^T Hx)^{1/2}$ and $(x^T Qx)^{1/2}$, respectively. In fact, it is easy to prove that the step size α_k^* defined in (3.8) satisfies $\frac{1}{2} \le \alpha_k^* \le \frac{m+1}{2}$.

Remark 5. For $i \in \{1, 2, \dots, m\}$, according to the optimality condition of the $\tilde{x}_i^k$-subproblem in (3.5a), there exists $f_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$ such that

(3.9)
$$(x_i - \tilde{x}_i^k)^T \left\{ f_i(\tilde{x}_i^k) - A_i^T \left[\lambda^k - \beta \left(\sum_{j=1}^i A\tilde{x}_j^k + \sum_{j=i+1}^m A_j x_j^k - b \right) \right] \right\} \ge 0 \ \ \forall\, x_i \in \mathcal{X}_i.$$

In the following we elucidate how to realize the proposed Gaussian back substitution procedure for some special cases of (1.3). The first special case of (1.3) is

$$\begin{array}{ll} \min & \sum_{i=1}^m \theta_i(x_i), \\ & \sum_{i=1}^m x_i = b, \\ & x_i \in \mathcal{X}_i \subset \Re^n, \quad i = 1, \dots, m, \end{array} \tag{3.10}$$

where all A_i's are identity matrices in (1.3). For this case, we have

$$A_i^T A_j = I_n \quad \forall\ i, j \in \{1, \dots, m\}.$$

Thus, for (3.10), the matrices M and H defined, respectively, in (3.1) and (3.3) reduce to

$$M = \begin{pmatrix} \beta I_n & 0 & \cdots & \cdots & 0 \\ \beta I_n & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & 0 \\ \beta I_n & \cdots & \beta I_n & \beta I_n & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{\beta} I_l \end{pmatrix}$$

and

$$H = \begin{pmatrix} \beta I_n & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & \beta I_n & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{\beta} I_l \end{pmatrix}.$$

Hence, the Gaussian back substitution procedure (3.5b) is characterized by the following system of linear equations:

$$\begin{pmatrix} x_2^{k+1} \\ \vdots \\ x_{m-1}^{k+1} \\ x_m^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x_2^{k} \\ \vdots \\ x_{m-1}^{k} \\ x_m^{k} \\ \lambda^{k} \end{pmatrix} + \alpha \begin{pmatrix} I_n & -I_n & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & I_n & -I_n & 0 \\ 0 & \cdots & 0 & I_n & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix} \begin{pmatrix} \tilde{x}_2^k - x_2^k \\ \vdots \\ \tilde{x}_{m-1}^k - x_{m-1}^k \\ \tilde{x}_m^k - x_m^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix}.$$

Therefore, the Gaussian back substitution procedure (3.5b) for the special case (3.10) can be completed by the following backtracking procedure:

$$\begin{cases} \lambda^{k+1} = \lambda^k + \alpha(\tilde{\lambda}^k - \lambda^k), \\ x_m^{k+1} = x_m^k + \alpha(\tilde{x}_m^k - x_m^k), \\ x_i^{k+1} = x_i^k + \alpha\{(\tilde{x}_i^k - x_i^k) - (\tilde{x}_{i+1}^k - x_{i+1}^k)\} \quad \text{for } i = m-1, \ldots, 2, \\ x_1^{k+1} = \tilde{x}_1^k. \end{cases}$$

Recall that we require $\alpha \in (0, 1)$ for ADM-G. Therefore, the value of α can be arbitrarily close to 1, and the asymptotical behavior (with $\alpha \to 1$) of the proposed ADM-G for (3.10) should coincide with the following scheme:

$$\text{(3.11)} \qquad \text{(Substitution form I)} \begin{cases} \lambda^{k+1} = \tilde{\lambda}^k, \\ x_m^{k+1} = \tilde{x}_m^k, \\ x_i^{k+1} = \tilde{x}_i^k - (\tilde{x}_{i+1}^k - x_{i+1}^k), \quad i = m-1, \ldots, 2, \\ x_1^{k+1} = \tilde{x}_1^k. \end{cases}$$

According to (3.11), for the special case of (3.10) with $m = 2$, the asymptotical behavior (with $\alpha \to 1$) of the proposed ADM-G reduces to the original ADM (1.2). Note that the back substitution procedure (3.11) differs slightly from the ASALM (1.4) only in the way of generating x_i^{k+1}'s ($i = 2, 3, \ldots, m-1$). In this sense, for the special case (3.10), the proposed ADM-G is asymptotically close to the ASALM (1.4) when $\alpha \to 1$.

Now, we consider the model

$$\text{(3.12)} \qquad \min \left\{ \sum_{i=1}^{m} \theta_i(x) \mid x \in \mathcal{X} \subset \Re^n \right\},$$

which has applications in many areas; see, e.g., [4, 17, 31]. Obviously, (3.12) can be easily reformulated as

$$\begin{array}{ll} \min & \sum_{i=1}^{m} \theta_i(x_i), \\ & x_1 = x_2 = \cdots = x_m, \\ & x_i \in \mathcal{X}, \quad i = 1, \ldots, m, \end{array} \tag{3.13}$$

which is a special case of the model (1.3) whose A_i matrices are, respectively, the ith column of the matrix

$$A = \begin{pmatrix} I_n & -I_n & 0 & \cdots & 0 \\ 0 & I_n & -I_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & I_n & -I_n \\ -I_n & 0 & \cdots & 0 & I_n \end{pmatrix}_{(mn)\times(mn)} \tag{3.14}$$

and $b = 0$. For such A_i's, we then have

$$A_i^T A_j = \begin{cases} 2I_n & \text{if } i = j, \\ -I_n & \text{if } |j - i| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, for (3.12), the matrices M and H defined, respectively, in (3.1) and (3.3), reduce to

$$M = \begin{pmatrix} 2\beta I_n & 0 & \cdots & \cdots & 0 \\ -\beta I_n & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & -\beta I_n & 2\beta I_n & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{\beta} I_l \end{pmatrix}$$

and

$$H = \begin{pmatrix} 2\beta I_n & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 2\beta I_n & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{\beta} I_l \end{pmatrix},$$

and we have

$$H^{-1}M^T = \begin{pmatrix} I_n & -\frac{1}{2} I_n & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & I_n & -\frac{1}{2} I_n & 0 \\ 0 & \cdots & 0 & I_n & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix}.$$

Hence, the Gaussian back substitution procedure (3.5b) for (3.12) is characterized by the following system of linear equations:

$$\begin{pmatrix} I_n & -\frac{1}{2}I_n & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & I_n & -\frac{1}{2}I_n & 0 \\ 0 & \cdots & 0 & I_n & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix} \begin{pmatrix} x_2^{k+1} - x_2^k \\ \vdots \\ x_{m-1}^{k+1} - x_{m-1}^k \\ x_m^{k+1} - x_m^k \\ \lambda^{k+1} - \lambda^k \end{pmatrix} = \alpha \begin{pmatrix} \tilde{x}_2^k - x_2^k \\ \vdots \\ \tilde{x}_{m-1}^k - x_{m-1}^k \\ \tilde{x}_m^k - x_m^k \\ \tilde{\lambda}^k - \lambda^k \end{pmatrix}.$$

Therefore, the Gaussian back substitution procedure (3.5b) for the special case (3.12) can be completed by the following backtracking procedure:

$$\begin{cases} \lambda^{k+1} = \lambda^k + \alpha(\tilde{\lambda}^k - \lambda^k), \\ x_m^{k+1} = x_m^k + \alpha(\tilde{x}_m^k - x_m^k), \\ (x_i^{k+1} - x_i^k) - \frac{1}{2}(x_{i+1}^{k+1} - x_{i+1}^k) = \alpha(\tilde{x}_i^k - x_i^k) \quad \text{for } i = m-1, \dots, 2, \\ x_1^{k+1} = \tilde{x}_1^k. \end{cases}$$

Similarly, the asymptotical behavior (with $\alpha \to 1$) of the proposed ADM-G should coincide with the following scheme. In particular, if $\alpha = 1$, the last scheme is reduced to

$$\text{(3.15)} \quad \text{(Substitution form II)} \begin{cases} \lambda^{k+1} = \tilde{\lambda}^k, \\ x_m^{k+1} = \tilde{x}_m^k, \\ x_i^{k+1} = \tilde{x}_i^k + \frac{1}{2}(x_{i+1}^{k+1} - x_{i+1}^k), \quad i = m-1, \dots, 2, \\ x_1^{k+1} = \tilde{x}_1^k. \end{cases}$$

Again, the scheme (3.15) is identical to the original ADM (1.2) for the special case of (3.12) with $m = 2$. Moreover, since the back substitution procedure (3.15) differs slightly from the ASALM (1.4) only in the way of generating x_i^{k+1}'s ($i = 2, 3, \dots, m-1$), the proposed ADM-G is asymptotically close to the ASALM (1.4) when $\alpha \to 1$ for solving (3.12).

4. Convergence of the ADM with Gaussian back substitution. In this section, we prove the convergence of the proposed ADM-G for solving (1.3). Our proof follows the analytic framework of contractive-type methods (see [2] for the definition), and it consists of the following three phases:

(1) Show that $-M^{-T}H(v^k - \tilde{v}^k)$ is a descent direction of the distance function $\frac{1}{2}\|v - v^*\|_G^2$ with $G = MH^{-1}M^T$ at the point $v = v^k$ whenever $\tilde{v}^k \neq v^k$.

(2) Show that the sequence generated by the proposed ADM-G is contractive with respect to $\mathcal{V}^*$.

(3) Prove the convergence of the proposed ADM-G.

Accordingly, we divide this section into three subsections to address the tasks listed above.

4.1. Verification of the descent direction. In this subsection, we mainly show that $-M^{-T}H(v^k - \tilde{v}^k)$ is a descent direction of the distance function $\frac{1}{2}\|v - v^*\|_G^2$ at the point $v = v^k$ whenever $\tilde{v}^k \neq v^k$, where $\tilde{v}^k$ is generated by the ADM step (3.5a), $v^* \in \mathcal{V}^*$, and $G = MH^{-1}M^T$. For this purpose, we first prove two lemmas.

LEMMA 4.1. *Let $\tilde{w}^k$ be generated by the ADM step* (3.5a) *from the given vector v^k. Let*

$$d_1(v^k, \tilde{v}^k) := \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ \beta A_2^T A_2 & 0 & \cdots & \cdots & 0 \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix} \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \tag{4.1}$$

and

$$d_2(v^k, \tilde{w}^k) := \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k \\ \sum_{i=1}^m A_i \tilde{x}_i^k - b \end{pmatrix} + \beta \begin{pmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_m^T \\ 0 \end{pmatrix} \left(\sum_{j=2}^m A_j (x_j^k - \tilde{x}_j^k) \right), \tag{4.2}$$

where $f_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$ satisfies (3.9). *Then, we have*

$$\tilde{w}^k \in \mathcal{W}, \qquad (w - \tilde{w}^k)^T \{ d_2(v^k, \tilde{w}^k) - d_1(v^k, \tilde{v}^k) \} \geq 0 \quad \forall\, w \in \mathcal{W}. \tag{4.3}$$

Proof. By using $\tilde{\lambda}^k = \lambda^k - \beta(\sum_{j=1}^m A_j \tilde{x}_j^k - b)$, it follows from (3.9) that

$$\tilde{x}_i^k \in \mathcal{X}_i,\ (x_i - \tilde{x}_i^k)^T \left\{ f_i(\tilde{x}_i^k) - A_i^T \tilde{\lambda}^k + \beta A_i^T \left(\sum_{j=i+1}^m A_j (x_j^k - \tilde{x}_j^k) \right) \right\} \geq 0 \ \forall\, x_i \in \mathcal{X}_i. \tag{4.4}$$

Thus, it follows from (4.4) that

$$\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_{m-1} - \tilde{x}_{m-1}^k \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \left\{ \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k \\ \vdots \\ f_{m-1}(\tilde{x}_{m-1}^k) - A_{m-1}^T \tilde{\lambda}^k \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k \end{pmatrix} + \beta \begin{pmatrix} A_1^T \left(\sum_{j=2}^m A_j (x_j^k - \tilde{x}_j^k) \right) \\ A_2^T \left(\sum_{j=3}^m A_j (x_j^k - \tilde{x}_j^k) \right) \\ \vdots \\ A_{m-1}^T \left(A_m (x_m^k - \tilde{x}_m^k) \right) \\ 0 \end{pmatrix} \right\} \geq 0 \tag{4.5}$$

is satisfied for all $x_i \in \mathcal{X}_i$ $(i = 1, 2, \ldots, m)$. Then, by adding the term

$$\beta \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_{m-1} - \tilde{x}_{m-1}^k \\ x_m - \tilde{x}_m^k \end{pmatrix}^T \begin{pmatrix} 0 \\ A_2^T \left(\sum_{j=2}^2 A_j (x_j^k - \tilde{x}_j^k) \right) \\ \vdots \\ A_{m-1}^T \left(\sum_{j=2}^{m-1} A_j (x_j^k - \tilde{x}_j^k) \right) \\ A_m^T \left(\sum_{j=2}^m A_j (x_j^k - \tilde{x}_j^k) \right) \end{pmatrix}$$

to both sides of (4.5), we get $\tilde{x}_i^k \in \mathcal{X}_i$ and

$$
\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_{m-1} - \tilde{x}_{m-1}^k \\ x_m - \tilde{x}_m^k \end{pmatrix}^T
\begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T\tilde{\lambda}^k + \beta A_1^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_2(\tilde{x}_2^k) - A_2^T\tilde{\lambda}^k + \beta A_2^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ f_{m-1}(\tilde{x}_{m-1}^k) - A_{m-1}^T\tilde{\lambda}^k + \beta A_{m-1}^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_m(\tilde{x}_m^k) - A_m^T\tilde{\lambda}^k + \beta A_m^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \end{pmatrix}
\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_{m-1} - \tilde{x}_{m-1}^k \\ x_m - \tilde{x}_m^k \end{pmatrix}^T
\begin{pmatrix} 0 \\ \beta A_2^T\big(\sum_{j=2}^2 A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ \beta A_{m-1}^T\big(\sum_{j=2}^{m-1} A_j(x_j^k - \tilde{x}_j^k)\big) \\ \beta A_m^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \end{pmatrix}
\quad \forall\, x_i \in \mathcal{X}_i, \;\; i = 1, \dots, m. \tag{4.6}
$$

Because $\sum_{j=1}^m A_j\tilde{x}_j^k - b = \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k)$, we have

$$
(\lambda - \tilde{\lambda}^k)^T\left(\sum_{j=1}^m A_j\tilde{x}_j^k - b\right) = (\lambda - \tilde{\lambda}^k)^T\frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k).
$$

Adding together (4.6) and the last equality, we get $\tilde{w}^k \in \mathcal{W}$ and

$$
\begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_{m-1} - \tilde{x}_{m-1}^k \\ x_m - \tilde{x}_m^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T
\begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T\tilde{\lambda}^k + \beta A_1^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_2(\tilde{x}_2^k) - A_2^T\tilde{\lambda}^k + \beta A_2^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ f_{m-1}(\tilde{x}_{m-1}^k) - A_{m-1}^T\tilde{\lambda}^k + \beta A_{m-1}^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ f_m(\tilde{x}_m^k) - A_m^T\tilde{\lambda}^k + \beta A_m^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \sum_{i=1}^m A_i\tilde{x}_i^k - b \end{pmatrix}
\geq \begin{pmatrix} x_1 - \tilde{x}_1^k \\ x_2 - \tilde{x}_2^k \\ \vdots \\ x_{m-1} - \tilde{x}_{m-1}^k \\ x_m - \tilde{x}_m^k \\ \lambda - \tilde{\lambda}^k \end{pmatrix}^T
\begin{pmatrix} 0 \\ \beta A_2^T\big(\sum_{j=2}^2 A_j(x_j^k - \tilde{x}_j^k)\big) \\ \vdots \\ \beta A_{m-1}^T\big(\sum_{j=2}^{m-1} A_j(x_j^k - \tilde{x}_j^k)\big) \\ \beta A_m^T\big(\sum_{j=2}^m A_j(x_j^k - \tilde{x}_j^k)\big) \\ \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) \end{pmatrix}
\quad \forall\, w \in \mathcal{W}.
$$

Recall the definitions of $d_1(v^k, \tilde{v}^k)$ and $d_2(v^k, \tilde{w}^k)$ in (4.1) and (4.2); the assertion (4.3) is proved. ∎

Remark 6. Note that $d_1(v^k, \tilde{v}^k)$ depends only on v^k and $\tilde{v}^k$, while $d_2(v^k, \tilde{w}^k)$ is determined by both v^k and $\tilde{w}^k$. The first row of the matrix associated with $d_1(v^k, \tilde{v}^k)$ is completely zero and seems redundant. But, we keep it for consistency of the dimensionality of $d_1(v^k, \tilde{v}^k)$ and $d_2(v^k, \tilde{w}^k)$.

LEMMA 4.2. *Let $\tilde{w}^k$ be generated by the ADM step* (3.5a) *from the given vector v^k, and let M be defined as in* (3.1)*. Then we have*

$$(4.7)\qquad (\tilde{v}^k - v^*)^T M (v^k - \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right) \quad \forall\, v^* \in \mathcal{V}^*.$$

Proof. Recall that $f_i(\tilde{x}_i^k) \in \partial\theta_i(\tilde{x}_i^k)$ satisfies (3.9); it follows from (4.3) that

$$(4.8)\qquad (\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \quad \forall w^* \in \mathcal{W}^*.$$

We now focus on the right-hand side of (4.8). First, for notational convenience, we denote

$$F(\tilde{w}^k) := \begin{pmatrix} f_1(\tilde{x}_1^k) - A_1^T \tilde{\lambda}^k \\ f_2(\tilde{x}_2^k) - A_2^T \tilde{\lambda}^k \\ \vdots \\ f_m(\tilde{x}_m^k) - A_m^T \tilde{\lambda}^k \\ \sum_{i=1}^m A_i \tilde{x}_i^k - b \end{pmatrix}.$$

Then, (4.2) can be rewritten as

$$d_2(v^k, \tilde{w}^k) = F(\tilde{w}^k) + \beta \begin{pmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_m^T \\ 0 \end{pmatrix} \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right).$$

Therefore, we have

$$(4.9)\qquad \begin{aligned} &(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \\ &\quad = \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right)^T \beta \left(\sum_{j=1}^{m} A_j (\tilde{x}_j^k - x_j^*) \right) + (\tilde{w}^k - w^*)^T F(\tilde{w}^k). \end{aligned}$$

Recall $\tilde{w} \in \mathcal{W}$ and use (2.2). Then it is easy to verify that

$$(\tilde{w}^k - w^*)^T F(\tilde{w}^k) \geq 0.$$

In addition, because

$$\sum_{j=1}^{m} A_j x_j^* = b \qquad \text{and} \qquad \beta \left(\sum_{j=1}^{m} A_j \tilde{x}_j^k - b \right) = \lambda^k - \tilde{\lambda}^k,$$

it follows from (4.9) that

$$(\tilde{w}^k - w^*)^T d_2(v^k, \tilde{w}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right) \quad \forall w^* \in \mathcal{W}^*. \tag{4.10}$$

Substituting (4.10) into (4.8), we obtain

$$(\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k) \geq (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right) \quad \forall w^* \in \mathcal{W}^*. \tag{4.11}$$

On the other hand, since (3.1) and (4.1), we have

$$d_1(v^k, \tilde{v}^k) = \begin{pmatrix} 0 \\ M(v^k - \tilde{v}^k) \end{pmatrix}, \tag{4.12}$$

which implies that

$$(\tilde{v}^k - v^*)^T M(v^k - \tilde{v}^k) = (\tilde{w}^k - w^*)^T d_1(v^k, \tilde{v}^k). \tag{4.13}$$

Therefore, the assertion (4.7) follows immediately from (4.11) and (4.13). ∎

Now, based on the last two lemmas, we are in position to prove a theorem which will be crucial for establishing the convergence of ADM-G.

Theorem 4.3. *Let $\tilde{v}^k$ be generated by the ADM procedure* (3.5a) *from the given vector v^k. Then, we have*

$$(v^k - v^*)^T M(v^k - \tilde{v}^k) \geq \frac{1}{2}\|v^k - \tilde{v}^k\|_H^2 + \frac{1}{2}\|v^k - \tilde{v}^k\|_Q^2 \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.14}$$

where M, H, and Q are defined as in (3.1), (3.3), *and* (3.2), *respectively.*

Proof. First, for all $v^* \in \mathcal{V}^*$, it follows from (4.7) that

$$\begin{aligned}(v^k - v^*)^T M(v^k - \tilde{v}^k) \geq\ & (v^k - \tilde{v}^k)^T M(v^k - \tilde{v}^k) \\ & + (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right).\end{aligned} \tag{4.15}$$

Now we treat the first term of the right-hand side of (4.15). Using the matrix M (see (3.1)), we have

$$\begin{aligned}&(v^k - \tilde{v}^k)^T M(v^k - \tilde{v}^k) \\ &= \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} \beta A_2^T A_2 & 0 & \cdots & \cdots & 0 \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{\beta} I_l \end{pmatrix} \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.\end{aligned} \tag{4.16}$$

Then, let us deal with the second term of the right-hand side of (4.15). By manipulations, we have

$$(4.17)\qquad (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right)$$

$$= \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ A_2 & A_3 & \dots & A_m & 0 \end{pmatrix} \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

Adding together (4.16) and (4.17), it follows that

$$(v^k - \tilde{v}^k)^T M (v^k - \tilde{v}^k) + (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right)$$

$$= \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix} \begin{pmatrix} \beta A_2^T A_2 & 0 & \cdots & \cdots & 0 \\ \beta A_3^T A_2 & \beta A_3^T A_3 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & \beta A_m^T A_m & 0 \\ A_2 & A_3 & \cdots & A_m & \frac{1}{\beta} I_l \end{pmatrix}^T \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}^T \begin{pmatrix} 2\beta A_2^T A_2 & \beta A_2^T A_3 & \cdots & \beta A_2^T A_m & A_2^T \\ \beta A_3^T A_2 & 2\beta A_3^T A_3 & \cdots & \beta A_3^T A_m & A_2^T \\ \vdots & \vdots & & \vdots & \vdots \\ \beta A_m^T A_2 & \beta A_m^T A_3 & \cdots & 2\beta A_m^T A_m & A_m^T \\ A_2 & A_3 & \cdots & A_m & \frac{2}{\beta} I_l \end{pmatrix} \begin{pmatrix} x_2^k - \tilde{x}_2^k \\ x_3^k - \tilde{x}_3^k \\ \vdots \\ x_m^k - \tilde{x}_m^k \\ \lambda^k - \tilde{\lambda}^k \end{pmatrix}.$$

Using the notation of the matrices H and Q in the right-hand side of the last equality, we obtain

$$(v^k - \tilde{v}^k)^T M (v^k - \tilde{v}^k) + (\lambda^k - \tilde{\lambda}^k)^T \left(\sum_{j=2}^{m} A_j (x_j^k - \tilde{x}_j^k) \right) = \frac{1}{2} \|v^k - \tilde{v}^k\|_H^2 + \frac{1}{2} \|v^k - \tilde{v}^k\|_Q^2.$$

Substituting the last equality in (4.15), the assertion (4.14) is proved. ∎

It follows from (4.14) that

$$\left\langle M H^{-1} M^T (v^k - v^*), M^{-T} H (\tilde{v}^k - v^k) \right\rangle \leq -\frac{1}{2} \|v^k - \tilde{v}^k\|_{(H+Q)}^2.$$

In other words, by setting

$$(4.18)\qquad G = M H^{-1} M^T,$$

$MH^{-1}M^T(v^k - v^*)$ is the gradient of the distance function $\frac{1}{2}\|v - v^*\|_G^2$, and $-M^{-T}H$ $(v^k - \tilde{v}^k)$ is a descent direction of $\frac{1}{2}\|v - v^*\|_G^2$ at the current point v^k whenever $\tilde{v}^k \neq v^k$.

4.2. The contractive property. In this subsection, we mainly prove that the sequence generated by the proposed ADM-G is contractive with respect to the set $\mathcal{V}^*$.

Recall that we follow the definition of a contractive-type method in the textbook [2]. With this contractive property, the convergence of the proposed ADM-G can be easily derived via standard analysis in the context of contraction methods.

THEOREM 4.4. *Let $\tilde{v}^k$ be generated by the ADM procedure* (3.5a) *from the given vector v^k. Let the matrix G be given by* (4.18). *For the new iterate v^{k+1} produced by the Gaussian back substitution* (3.6), *there exists a constant $c_0 > 0$ such that*

$$\|v^{k+1} - v^*\|_G^2 \le \|v^k - v^*\|_G^2 - c_0\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) \quad \forall\, v^* \in \mathcal{V}^*, \tag{4.19}$$

where H and Q are defined as in (3.3) *and* (3.2), *respectively. That is, the sequence $\{v^k\}$ is G-norm contractive with respect to $\mathcal{V}^*$.*

Proof. For $G = MH^{-1}M^T$ and any $\alpha \ge 0$, we obtain

$$\begin{aligned} &\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \\ &= \|v^k - v^*\|_G^2 - \|(v^k - v^*) - \alpha M^{-T}H(v^k - \tilde{v}^k)\|_G^2 \\ &= 2\alpha(v^k - v^*)^T M(v^k - \tilde{v}^k) - \alpha^2\|v^k - \tilde{v}^k\|_H^2. \end{aligned} \tag{4.20}$$

Substituting (4.14) into the right-hand side of the last equation, we get

$$\begin{aligned} &\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \\ &\ge \alpha\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) - \alpha^2\|v^k - \tilde{v}^k\|_H^2 \\ &= \alpha(1-\alpha)\|v^k - \tilde{v}^k\|_H^2 + \alpha\|v^k - \tilde{v}^k\|_Q^2, \end{aligned}$$

and thus

$$\begin{aligned} &\|v^{k+1} - v^*\|_G^2 \\ &\quad \le \|v^k - v^*\|_G^2 - \alpha\big((1-\alpha)\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) \quad \forall\, v^* \in \mathcal{V}^*. \end{aligned} \tag{4.21}$$

Set $c_0 = \alpha(1-\alpha)$. Recall that $\alpha \in (0,1)$. Thus the assertion is proved. ∎

Based on the assertion (4.19), some properties of the sequence $\{v^k\}$ can be immediately derived, and we summarize them in the following corollary.

COROLLARY 4.5. *Let $\tilde{v}^k$ be generated by the ADM procedure* (3.5a) *from the given vector v^k. Then*

(1) *the sequence $\{v^k\}$ is bounded;*

(2) $\lim_{k\to\infty} \|A_i(x_i^k - \tilde{x}_i^k)\| = 0$ *for $i = 2, \ldots, m$, and* $\lim_{k\to\infty} \|\lambda^k - \tilde{\lambda}^k\| = 0$.

Proof. The first property is an obvious fact based on (4.19). For the second property, it follows from (4.19) that

$$\sum_{k=0}^{\infty} c_0\|v^k - \tilde{v}^k\|_H^2 \le \|v^0 - v^*\|_G^2,$$

which implies that $\lim_{k\to\infty} \|v^k - \tilde{v}^k\|_H^2 = 0$. Thus, the second property is proved. ∎

COROLLARY 4.6. *The assertion of Theorem* 4.4 *also holds if the Gaussian back substitution is* (3.7).

Proof. Analogous to the proof of Theorem 4.4, we have that

$$\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \ge 2\gamma\alpha_k^*(v^k - v^*)^T M(v^k - \tilde{v}^k) - (\gamma\alpha_k^*)^2\|v^k - \tilde{v}^k\|_H^2, \tag{4.22}$$

where α_k^* is given by (3.8). According to (3.8), we have that

$$\alpha_k^*\|v^k - \tilde{v}^k\|_H^2 = \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big).$$

Then, it follows from the above equality and (4.14) that

$$\begin{aligned}&\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2\\ &\geq \gamma\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) - \frac{1}{2}\gamma^2\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big)\\ &= \frac{1}{2}\gamma(2-\gamma)\alpha_k^*\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big).\end{aligned}$$

Because $\alpha_k^* \geq \frac{1}{2}$, it follows from the last inequality that

$$\begin{aligned}&\|v^{k+1} - v^*\|_G^2\\ &\quad\leq \|v^k - v^*\|_G^2 - \frac{1}{4}\gamma(2-\gamma)\big(\|v^k - \tilde{v}^k\|_H^2 + \|v^k - \tilde{v}^k\|_Q^2\big) \quad \forall\, v^* \in \mathcal{V}^*.\end{aligned} \tag{4.23}$$

Since $\gamma \in (0, 2)$, the assertion of this corollary follows directly from (4.23). ☐

4.3. Convergence. The proved lemmas and theorems are adequate for establishing the global convergence of the proposed ADM-G, and the analytic framework is standard in the context of contractive-type methods.

THEOREM 4.7. *Let $\{w^k\}$ be the sequence generated by the proposed ADM-G. Then, $\{w^k\}$ converges to a solution point of* (2.2).

Proof. First of all, since it is bounded, the sequence $\{v^k\}$ has at least one cluster point, and we denote it by $v^\infty = (x_2^\infty, x_3^\infty, \ldots, x_m^\infty, \lambda^\infty) \in \mathcal{V}$. In addition, let $\{v^{k_j}\}$ be the subsequence converging to v^∞. Obviously, it follows from (3.5a) that

$$A_1 x_1^{k+1} = A_1 \tilde{x}_1^k = \frac{1}{\beta}(\lambda^k - \tilde{\lambda}^k) - \left(\sum_{j=2}^m A_j \tilde{x}_j^k - b\right).$$

Therefore, assertion (2) of Corollary 4.5 implies that the sequence $\{x_1^k\} \subset \mathcal{X}_1$ generated by ADM-G is also bounded. We denote by x_1^∞ a cluster point and let $\{x_1^{k_j}\}$ be the subsequence converging to x_1^∞. Then, the sequence $\{w^k\}$ generated by the proposed algorithm is bounded. Moreover, $w^\infty = (x_1^\infty, v^\infty) \in \mathcal{W}$ is a cluster point of $\{w^k\}$, and $w^{k_j} = (x_1^{k_j}, v^{k_j})$ is the subsequence converging to w^∞.

Now, we show that w^∞ is a solution of (2.2). By taking the limit over j in (3.9) (see also (4.4)) and using assertion (2) in Corollary 4.5, we have that there exists $f_i(x_i^\infty) \in \partial\theta_i(x_i^\infty)$ such that

$$(x_i - x_i^\infty)^T\big\{f_i(x_i^\infty) - A_i^T\lambda^\infty\big\} \geq 0 \quad \forall\, x_i \in \mathcal{X}_i, \quad i = 1, 2, \ldots, m. \tag{4.24}$$

In addition, it follows from (3.5a) and assertion (2) of Corollary 4.5 that

$$\sum_{j=1}^m A_j x_j^\infty - b = 0. \tag{4.25}$$

Thus, (4.24) and (4.25) imply that

$$w^\infty \in \mathcal{W},\ (w - w^\infty)^T \begin{pmatrix} f_1(x_1^\infty) - A_1^T\lambda^\infty \\ f_2(x_2^\infty) - A_2^T\lambda^\infty \\ \vdots \\ f_m(x_m^\infty) - A_m^T\lambda^\infty \\ \sum_{j=1}^m A_j x_j^\infty - b \end{pmatrix} \geq 0 \quad \forall\, w \in \mathcal{W},\ i = 1, \ldots, m. \tag{4.26}$$

Therefore, w^∞ is a solution point of (2.2).

On the other hand, taking v^* in (4.19) as v^∞, it is obvious that the sequence $\{\|v^k - v^\infty\|\}$ is nonincreasing. Therefore, the sequence $\{v^k\}$ converges to v^∞. Accordingly, the sequence $\{w^k\}$ converges to w^∞, which is a solution of (2.2). ☐

5. Numerical results. In this section, we apply the proposed ADM-G to solve several concrete applications of the model (1.3) arising in different areas and report the numerical results. To further verify its numerical efficiency, we also compare the proposed ADM-G numerically with some customized efficient methods for these applications. As we have mentioned, for a concrete application of the abstract model (1.3), the involved θ_i functions and the coefficient A_i matrices usually have particular properties, and thus the decomposed subproblems (3.5a) are often easy enough to have closed-form solutions or can be easily solved up to high precision. As we shall show, this fact contributes much to the numerical efficiency of the proposed ADM-G.

All code was written in MATLAB v7.1 (R14) and performed on a T6500 notebook equipped with Windows XP, 2.1 GHz IntelCore 2 Duo CPU, and 2GB of memory.

5.1. Recovering low-rank and sparse components of matrices. In this subsection, we apply the proposed ADM-G to solve the problem of recovering low-rank and sparse components of matrices from incomplete and noisy observation, which was launched in [40]. More specifically, the model is

$$\begin{array}{rl} \min_{L,S} & \|L\|_* + \tau\|S\|_1 \\ \text{subject to} & \|P_\Omega(C - L - S)\|_F \le \delta, \end{array} \tag{5.1}$$

where $C \in \mathcal{R}^{l\times n}$ is a given matrix (data); $\|\cdot\|_*$ denotes the nuclear norm (defined as the sum of all singular values) which aims at inducing the low-rank component $L \in \mathcal{R}^{l\times n}$ (see, e.g., [36]); $\|\cdot\|_1$ (defined as the sum of the absolute values of all entries) is to induce the sparse component S; $\tau > 0$ is a constant balancing the low-rank and sparsity; Ω is a subset of the index set $\{1, 2, \dots, l\} \times \{1, 2, \dots, n\}$, and we assume that only those entries $\{C_{ij}, (i, j) \in \Omega\}$ can be observed, the incomplete observation information is summarized by the operator $P_\Omega : \mathcal{R}^{l\times n} \to \mathcal{R}^{l\times n}$, which is the orthogonal projection onto the span of matrices vanishing outside of Ω so that the ijth entry of $P_\Omega(X)$ is X_{ij} if $(i, j) \in \Omega$ and zero otherwise; $\delta > 0$ is the Gaussian noise level; and $\|\cdot\|_F$ denotes the standard Frobenius norm. Note that (5.1) is a generalized model of the matrix decomposition problem in [7] and the robust principal component analysis model in [5].

As analyzed in [40], by introducing $M := P_\Omega(C)$, (5.1) can be rewritten as

$$\begin{array}{rl} \min_{L,S,Z} & \|L\|_* + \tau\|S\|_1 \\ \text{subject to} & L + S + Z = M, \\ & Z \in \mathbf{B} := \{Z \in \mathcal{R}^{l\times n} \mid \|P_\Omega(Z)\|_F \le \delta\}, \end{array} \tag{5.2}$$

which is a special case of the model (1.3) (actually (3.10)) but with matrix variables (as we have mentioned, the proposed method is still applicable for the extension of (3.10) with matrix variables). Note that the substitution form (3.11) is applicable for (5.2).

In [40], the ASALM (1.4) was shown to perform very efficiently, even though its convergence is still ambiguous. Due to this lack of convergence, a variant of ASALM (denoted by VASALM) with convergence was also proposed in [40]. But, according to the numerical results in [40], empirically VASALM is less efficient than ASALM. In the following, we shall compare the proposed ADM-G with both ASALM and VASALM for solving (5.2).

We generate the data of (5.1) randomly in the same way as [40]. More specifically, let $C = L^* + S^*$. The low-rank matrix L^* is generated by $L^* = UR^T$, where U and R are independent $l \times r$ matrices whose entries are independently and identically (i.i.d.) Gaussian random variables with zero mean and unit variance. Hence, the rank of L^* is `r`. The index of the observed entries, i.e., Ω, is determined randomly. The support $\Gamma \subset \Omega$ of the sparse matrix S^* is chosen uniformly and randomly, and the nonzero entries of S^* are i.i.d. uniformly in the interval $[-500, 500]$ (thus, the nonzero entries of S^* can be large). Let `sr`, `spr`, and `rr` represent the ratios of sample (observed) entries (i.e., $|\Omega|/ln$), the number of nonzero entries of S^* (i.e., $\|S^*\|_0/ln$), and the rank of L^* (i.e., $r/\min(l,n)$), respectively. In our experiments, we choose $l = n = 500$; $\texttt{sr} = 0.8$, and we set $\tau = 1/\sqrt{n}$ in (5.1). We test some scenarios of `rr` and `spr`. The value δ is chosen as $\delta = \sqrt{n + \sqrt{8n}\sigma}$.

As in [40], the stopping criterion is set as

$$\text{RelChg} := \frac{\|(L^{k+1}, S^{k+1}) - (L^k, S^k)\|_F}{\|(L^k, S^k)\|_F + 1} \le Tol, \tag{5.3}$$

where Tol is the tolerance of the relative errors of the recovered low-rank and sparse components. Our numerical experiments focus on the cases of $\sigma = 0$ (i.e., the Gaussian noiseless case) and $\sigma = 10^{-3}$ (i.e., the Gaussian noise case). For the case of $\sigma = 0$, the tolerance in (5.3) is set as $1e-5$, and for the case of $\sigma = 10^{-3}$, the tolerance in (5.3) is set as 0.5σ. For all of the implemented methods and tested scenarios, the value of β is determined simply by

$$\beta = \begin{cases} 0.1\frac{|\Omega|}{\|P_\Omega(C)\|_1} & \text{if } \texttt{spr} = 0.05, \\ 0.15\frac{|\Omega|}{\|P_\Omega(C)\|_1} & \text{if } \texttt{spr} = 0.1, \end{cases} \tag{5.4}$$

and the initial iterate is $(L^0, S^0, Z^0) = (\mathbf{0}, \mathbf{0}, \mathbf{0})$. Since α can be arbitrarily close to 1, empirically we take $\alpha = 1$ for ADM-G; i.e., we use the substitution scheme (3.11). For the parameter μ required by VASALM, we set it as 1.5. We denote by $(\hat{L}, \hat{S})$ the iterate when the stopping criterion (5.3) is achieved.

In Table 5.1, we report the numerical results of VASALM, ASALM, and ADM-G. More specifically, for different methods, we report the relative error of the recovered sparse component ($ErrsSP := \frac{\|\hat{S}-S^*\|_F}{\|S^*\|_F}$), the relative error of the recovered low-rank component ($ErrsLR := \frac{\|\hat{L}-L^*\|_F}{\|L^*\|_F}$), the computing time in seconds ("Time(s)") and the number of eigenvalue decompositions required by the L-related subproblems ("#Eig").

According to the data in Table 5.1, as we expect, ASALM still performs the best among the tested methods in terms of both accuracy and speed. Nevertheless, the proposed ADM-G has better numerical performance than VASALM and it is very competitive with ASALM. As we have emphasized, the convergence of ASALM (i.e., (1.4)) is still ambiguous, while the convergence of ADM-G has been established in this paper. Therefore, the proposed ADM-G with favorable numerical performance and proved convergence can be used as a surrogate of ASALM whenever the latter is efficient.

To see the comparison clearly, we focus on the particular case where $l = n = 500$, $\texttt{spr} = 0.05$, $\texttt{rr} = 0.05$, $\texttt{sr} = 0.7$, and $\sigma = \texttt{1e} - 3$, and we visualize the iterative processes of different methods in Figure 5.1. More specifically, we plot the evolutions of the rank of the recovered components, the relative error $ErrsSP$, and the relative error $ErrsLR$, with respect to the iterations.

TABLE 5.1
Numerical comparison of VASALM, ASALM, and ADM-G for (5.1).

rr	spr	$\frac{\\|\hat{S}-S^*\\|_F}{\\|S^*\\|_F}$			$\frac{\\|\hat{L}-L^*\\|_F}{\\|L^*\\|_F}$			Time(s)			#Eig		
		VASALM	ASALM	ADM-G	VASALM	ASALM	ADM-G	VASALM	ASALM	ADM-G	VASALM	ASALM	ADM-G
$l = n = 500$, sr $= 0.8$, and $\sigma = 0$													
0.05	0.05	5.33e-6	3.18e-6	3.00e-5	1.96e-4	2.47e-5	1.66e-4	33.6	18.4	20.5	38	21	23
	0.1	1.10e-5	9.22e-6	2.78e-5	2.65e-4	8.83e-5	2.63e-4	33.9	19.9	21.5	38	22	24
0.1	0.05	2.44e-5	3.62e-6	3.29e-5	2.17e-4	3.94e-5	1.90e-4	34.9	21.3	25.8	44	26	31
	0.1	7.68e-5	1.76e-5	4.20e-5	7.21e-4	1.73e-4	3.64e-4	31.5	23.4	29.8	36	26	32
$l = n = 500$, sr $= 0.8$ and $\sigma = 1e-3$													
0.05	0.05	2.26e-4	1.52e-4	3.20e-4	8.11e-3	1.24e-3	3.88e-3	14.6	10.0	10.1	18	12	12
	0.1	3.37e-4	1.59e-4	3.43e-4	1.27e-2	2.41e-3	5.86e-3	13.7	10.0	11.0	17	12	13
0.1	0.05	3.32e-4	1.61e-4	3.97e-4	8.82e-3	2.37e-3	4.09e-3	17.6	11.5	14.4	22	14	17
	0.1	4.03e-4	2.46e-4	5.19e-4	8.55e-3	4.17e-3	6.73e-3	15.4	12.5	15.2	18	14	17

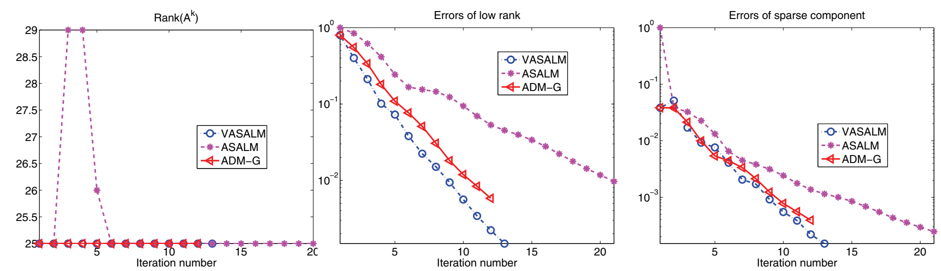

FIG. 5.1. *Evolution of rank (left), relative errors of low-rank (middle), and sparse components (right).*

According to the curves in Figure 5.1, ADM-G performs as well as ASALM in terms of the rank evolution of the recovered low-rank components. In terms of the relative error evolutions of both recovered low-rank components and sparse components, the performance of ADM-G is slightly worse than that of ASALM, but it is still much better than that of VASALM.

5.2. Fermat–Weber problem. In this subsection, our purpose is to compare the proposed ADM-G with the fast multiple splitting algorithm (FaMSA) proposed in [17]. Thus, we consider the same Fermat–Weber problem as in [17],

$$\min F(x) = \sum_{i=1}^{m} \|x - c^i\|, \tag{5.5}$$

where $c^i \in \mathcal{R}^n (i = 1, \ldots, m)$ are given points. Note that (5.5) is a special case of (3.12) and can be reformulated as

$$\begin{aligned} \min \sum_{i=1}^{m} \|x_i - c^i\|, \\ x_1 = x_2 = \cdots = x_m. \end{aligned}$$

Therefore, the proposed ADM-G is applicable.

We generate the c^i vectors exactly in the way suggested in [17], i.e., the c^i's are i.i.d. Gaussian entries from $\mathcal{N}(0, n)$. To implement FaMSA, we take the recommended values in [17] for all parameters of FaMSA. To implement ADM-G, we take $\beta = 0.01\frac{\sum_{i=1}^m \sum_{j=1}^n |c_j^i|}{mn}$ and again set $\alpha = 1$ in the Gaussian back substitution (i.e., we

TABLE 5.2
Comparison of ADM-G and FaMSA in [17].

Problem		It.		Time(s)		Obj-End	
m	n	ADM-G	FaMSA	ADM-G	FaMSA	ADM-G	FaMSA
50	50	64	63	0.14	0.98	1.736863380e4	1.736863537e4
100	100	154	120	0.59	12.89	9.788833117e4	9.788836287e4
200	200	238	163	3.13	65.75	5.608766247e5	5.608770174e5
250	250	330	155	6.84	118.59	9.818182759e5	9.818192518e5

use the substitution scheme (3.15)). Both methods start with the initial iterate $x_i = \mathbf{0}$ $(i = 2, \ldots, m)$ and $\lambda_i = \mathbf{0}$ $(i = 1, \ldots, m)$.

The authors of [17] suggested solving the second-order cone programming (SOCP) reformulation of (5.5) first (e.g., by the Mosek package [30]) to obtain the solution x^*, and then using the following criterion to implement FaMSA:

$$RelErr := \frac{|\min_{i=1,2,\ldots,m}\{F(x_i^k)\} - F(x^*)|}{F(x^*)} < 10^{-6}. \tag{5.6}$$

At the same time, as mentioned in [17], for large-scale cases of (5.5), solving (5.5) a priori based on its SOCP reformulation is either impossible or prohibitively expensive, as the dimensionality of SOCP reformulation is significantly enlarged. For these concerns, we choose the following stopping criterion to implement FaMSA:

$$RelErr := \frac{|\min_{i=1,2,\ldots,m}\{F(x_i^k)\} - \min_{i=1,2,\ldots,m}\{F(x_i^{k-1})\}|}{|\min_{i=1,2,\ldots,m}\{F(x_i^{k-1})\}|} < 10^{-8}.$$

To implement ADM-G, our stopping criterion is

$$\max\left\{\frac{\max_i \|x_i^{k+1} - x_i^k\|_1}{\|x_i^1 - x_i^0\|_1}, \frac{\|\lambda^{k+1} - \lambda^k\|_1}{\|\lambda^1 - \lambda^0\|_1}\right\} < 10^{-4}.$$

In Table 5.2, for some cases of m and n, we report the iteration numbers ("It.") and computing times in seconds ("Time(s)") for the proposed ADM-G and FaMSA. As these two methods adopt different stopping criteria, for fair comparison we also report the objective function values of (5.5) ("Obj-End") when these two methods are terminated.

Note that the computation load per iteration of FaMSA is much more than that of ADM-G ($O(m(m-1)n)$ versus $O(mn)$). Therefore, to obtain the same (or slightly different) objective function values, the computing time of ADM-G has to be significantly less than that of FaMSA even though it requires more iterations. The data in Table 5.2 supports the efficiency of the proposed ADM-G for solving (5.5).

5.3. Super-resolution from a sequence of low-resolution frames. In this subsection, we apply the proposed ADM-G to solve the super-resolution problem from a sequence of low-resolution frames arising in the discipline of image processing, which can be reformulated as a special case of (1.3). By this application, we shall demonstrate that the proposed ADM-G performs as well as the ASALM (i.e, (1.4)) in the sense that both require the same number of iterations for reconstructing the image with the same quality. This application thus further verifies the efficiency of the proposed ADM-G.

For the background of the super-resolution imaging problem, we refer the reader to, e.g., [3, 27]. Here we provide a brief review. An image acquisition system composed of an array of sensors, where each sensor has a subarray of sensing elements of suitable size, has recently been popular for increasing the spatial resolution with high signal-to-noise ratio (SNR) beyond the performance bound of technologies that constrain the manufacture of imaging devices. The multiframe-based super-resolution problem is to reconstruct a high-resolution image (HRI) from a sequence of low-resolution images (LRIs) about a scene (which may be blurred and noised), and the aliasing effects in LRIs (e.g., subtle shift or shake between LRIs) enable the possibility of reconstruction. Note that the LRIs can be yielded when an HRI is blurred, noised, motioned, and down-sampled. Thus, we have

$$b_i = RS_iG_iu + \mathbf{n}_i, \quad i = 1, \ldots, p, \tag{5.7}$$

where $b_i \in \mathcal{R}^l$ $(i = 1, \ldots, k)$ are observed frames of LRIs, $R \in \mathcal{R}^{l\times n}$ is a down-sampling operator, $S_i \in \mathcal{R}^{n\times n}$ are motioning operators, $G_i \in \mathcal{R}^{n\times n}$ are blurring operators, $\mathbf{n}_i \in \mathcal{R}^l$ are noise to individual LRIs, and $u \in \mathcal{R}^n$ is the HRI to be reconstructed.

The inverse problem (5.7) is usually ill-posed and computationally challenging. Therefore, it is not practical to solve it directly, and certain regularization techniques are required. One of the most popular regularization techniques is the total variation (TV) regularization, proposed in the seminal work [37], mainly because of its capability of preserving the edges of images. More specifically, let $\partial_1 : \mathcal{R}^n \to \mathcal{R}^n$ and $\partial_2 : \mathcal{R}^n \to \mathcal{R}^n$ be the finite-difference operators in the horizontal and vertical directions, respectively, and let $\nabla := (\partial_1, \partial_2)$ denote the gradient operator. The TV model for reconstructing the HRI from a sequence of LRIs can be formulated as

$$\min \quad \left\{ \frac{1}{2}\sum_{i=1}^{p} \|RS_iG_iu - b_j\|^2 + \tau\|\nabla u\|_1 \;\middle|\; u \in \Omega \right\}, \tag{5.8}$$

where $\Omega := \{u \in \mathcal{R}^n \mid 0 \le u \le 255\}$, $\tau > 0$ is a constant balancing the data-fitting and regularization, and $\|\cdot\|_1$ defined on $\mathcal{R}^n \times \mathcal{R}^n$ is given by

$$\||y|\|_1 := \|(|y|)\|_1 \quad \forall y = (y_1, y_2) \in \mathcal{R}^n \times \mathcal{R}^n.$$

Here, $|y| := \sqrt{y_1^2 + y_2^2} \in \mathcal{R}^n$ is understood in the componentwise sense: $(|y|)_i := \sqrt{(y_1)_i^2 + (y_2)_i^2}$ (see, e.g., [35, Chapter 1]). However, the model (5.8) is not easy due to the nonsmoothness of the term $\|\cdot\|_1$, the high dimensionality, and the ill-posedness. Here, we focus on the case that $G_i = I$; i.e., there is no blur on the HRI, and the motion operator is known. Thus, we consider the following model for reconstructing the HRI from a sequence of LRIs:

$$\min \quad \left\{ \frac{1}{2}\sum_{i=1}^{p} \|RS_iu - b_i\|^2 + \tau\|\nabla u\|_1 \;\middle|\; u \in \Omega \right\}. \tag{5.9}$$

Note that the model (5.9) differs from the super-resolution model in [10] in that the additional constraint Ω is required for u. In fact, as [29] shows, this additional constraint Ω often results in a reconstructed image with much higher quality.

We now illustrate that the model (5.9) can be reformulated as a special case of (1.3). In fact, introducing the auxiliary variables x_i and letting $x_i = S_iu$, the model (5.9) can be reformulated as

$$\min \left\{ \frac{1}{2}\sum_{i=1}^{p} \|Rx_i - b_i\|^2 + \tau\|y\|_1 \;\middle|\; x_i = S_iu, i = 1, \ldots, p; y = \nabla u, u \in \Omega \right\}. \tag{5.10}$$

We further rename y, u in (5.10) as x_{p+1}, x_{p+2}, and then (5.10) can be rewritten as

$$\min\left\{\frac{1}{2}\sum_{i=1}^{p}\|Rx_i-b_i\|^2+\tau\|x_{p+1}\|_1 \;\middle|\; x_i=S_ix_{p+2}, i=1,\dots,p; x_{p+1} = \nabla x_{p+2}, x_{p+2}\in\Omega\right\}, \tag{5.11}$$

which is obviously a special case of (1.3) with $m=p+2$, $b=0$:

$$\theta_i(x_i)=\begin{cases}\frac{1}{2}\|Rx_i-b_i\|^2 & \text{if } i=1,\dots,p,\\ \tau\|\nabla x_i\|_1 & \text{if } i=p+1,\\ 0 & \text{if } i=p+2;\end{cases} \tag{5.12}$$

$$A_1=\begin{pmatrix} I\\ 0\\ \vdots\\ \vdots\\ 0\end{pmatrix},\dots,A_i=\begin{pmatrix} 0\\ \vdots\\ I\\ \vdots\\ 0\end{pmatrix},\dots,A_{p+1}=\begin{pmatrix} 0\\ 0\\ \vdots\\ \vdots\\ I\end{pmatrix},\ A_{p+2}=\begin{pmatrix} -S_1\\ -S_2\\ \vdots\\ -S_p\\ -\nabla\end{pmatrix}; \tag{5.13}$$

and

$$\mathcal{X}_i=\begin{cases}\mathcal{R}^n & \text{if } i=1,\dots,p+1,\\ \Omega & \text{if } i=p+2.\end{cases} \tag{5.14}$$

According to (5.13), it is easy to verify that

$$A_i^TA_j=\begin{cases} I_n & \text{if } i=j,\ \ i,j\in\{1,\dots,p\},\\ 0 & \text{if } i\neq j,\ \ i,j\in\{1,\dots,p\},\\ -S_j, & i=p+2,\ j\in\{1,\dots,p\},\\ -S_i, & j=p+2,\ i\in\{1,\dots,p\},\\ -\nabla, & i\neq j,\ i,j\in\{p+1,p+2\},\\ \sum_{j=1}^{p}S_j^TS_j+\nabla^T\nabla, & i=j=p+2.\end{cases}$$

Therefore, for the model (5.11), the matrix $H^{-1}M^T$ defined in (3.4) for performing the Gaussian back substitution reduces to

$$H^{-1}M^T=\begin{pmatrix} I_n & 0 & \cdots & -S_2 & 0\\ 0 & \ddots & \ddots & \vdots & \vdots\\ \vdots & \ddots & \ddots & -\nabla & 0\\ 0 & \cdots & 0 & I_n & 0\\ 0 & \cdots & 0 & 0 & I_{(p+1)n}\end{pmatrix}.$$

Accordingly, by taking $\alpha = 1$, the Gaussian back substitution step (3.5b) yields the $(k+1)$th iteration by

$$\left\{\begin{array}{ll} \lambda^{k+1} = \tilde{\lambda}^k, \\ x_{p+2}^{k+1} = \tilde{x}_{p+2}^k, \\ x_i^{k+1} = \tilde{x}_i^k + S_i(x_{p+2}^{k+1} - x_{p+2}^k), \quad i = 2, \ldots, p, \\ x_{p+1}^{k+1} = \tilde{x}_{p+1}^k + (\nabla x_{p+2}^{k+1} - \nabla x_{p+2}^k), \\ x_1^{k+1} = \tilde{x}_1^k. \end{array}\right. \tag{5.15}$$

On the other hand, based on (5.12)–(5.14), it is easy to verify that in order to generate the $(k+1)$th iteration, the ADM procedure (3.5a) for (5.11) reduces to the following tasks:

$$\left\{\begin{array}{l} \tilde{x}_i^k = (R^T R + \beta I)^{-1}(R^T b_i + \lambda_i^k + \beta S_i x_{p+2}^k), \quad i = 1, \ldots, p, \\ \tilde{x}_{p+1}^k = \arg\min \tau \|x_{p+1}\| + \frac{\beta}{2}\|x_{p+1} - \nabla x_{p+2}^k - \frac{\delta^k}{\beta}\|^2, \\ \tilde{x}_{p+2}^k = \arg\min_{x_{p+2}\in\Omega} \left\{ \begin{array}{c} \frac{\beta}{2}\|\nabla x_{p+2} - \tilde{x}_{p+1}^k\|^2 + \sum_{j=1}^p \langle \lambda_j^k, S_j x_{p+2}\rangle \\ +\langle \delta^k, \nabla x_{p+2}\rangle + \frac{\beta}{2}\sum_{j=1}^p \|\tilde{x}_j^k - S_j x_{p+2}\|^2 \end{array} \right\}, \\ \tilde{\lambda}_j^k = \lambda_j^k - \beta(\tilde{x}_j^k - S_j \tilde{x}_{p+1}^k), \quad j = 1, \ldots, p, \\ \tilde{\delta}^k = \delta^k - \beta(\tilde{x}_{p+1}^k - \nabla \tilde{x}_{p+2}^k), \end{array}\right. \tag{5.16}$$

where δ_k is also a Lagrange multiplier. Note that the ADM procedure (5.16) of the next iteration relies only on x_{p+2}^{k+1} and λ^{k+1} generated by the last iteration, while the back substitution step (5.15) shows that the proposed ADM-G makes no difference from the ASALM (i.e., (1.4)) in x_{p+2}^{k+1} and λ^{k+1}. Therefore, although it requires an additional back substitution step, the proposed ADM-G actually generates the same input for the ADM procedure (5.16) as ASALM does. For this reason, these two methods require the same numbers of iterations to satisfy a given stopping criterion, and they differ only in the last iteration, which is caused by (5.15). In addition, since the variable x_{p+2} (i.e., u in (5.10)) denotes the image to be reconstructed, (5.15) indicates that these two methods actually reconstruct the same images. Overall, the proposed ADM-G performs as well as ASALM: reconstructing the same images with the same number of iterations. But, the computation time of ADM-G is expected to be longer due to the additional computation in (5.15). We will verify it numerically.

For the numerical experiment, we choose the images of "MRI" (128×128) and "montage.png" (256×256), shown in Figure 5.2. Both of the original HRIs are degraded by the motion operators proposed in [10] and noised by the Gaussian noise with 0.01 standard deviation and zero mean. Then, we down-sampled the degraded HRIs with various factors to obtain 32 LRI frames. Therefore, the tested problems are special cases of (1.3) with $p = 32$. Some frames of LRIs are shown in Figure 5.3.

In our numerical experiments, we set $\tau = 1e-3$ in (5.9) and $\beta = 5e-3$ for both ADM-G and ASALM. The initial point x_{p+2}^0 is set as $\mathbf{0}$. We apply the method in [1] to solve the x_{p+2}-related subproblem in (5.16), and the iterative number for the inner iteration is set as 10. The stopping criterion is

$$\text{RelChg} := \frac{\|x_{p+2}^{k+1} - x_{p+2}^k\|}{\|x_{p+2}^k\|} \leq 1e-3. \tag{5.17}$$

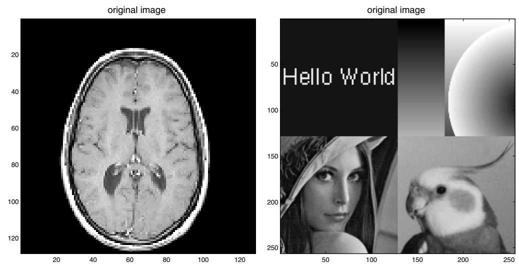

Fig. 5.2. *Original images: MRI (left) and montage.png (right).*

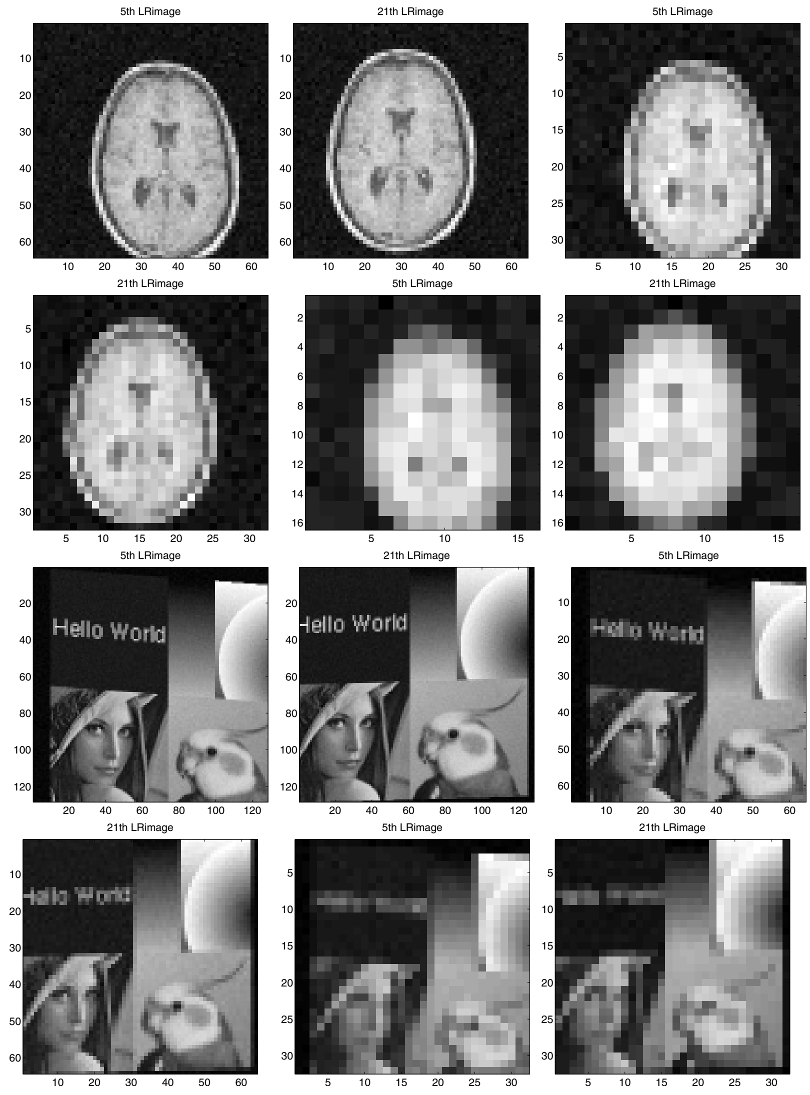

Fig. 5.3. *Frames of LRIs with various down-sampling factors s. Top two rows: LRIs of MRI with $s = 2$, 4, 8. Bottom two rows: LRIs of montage.png with $s = 2$, 4, 8.*

TABLE 5.3
Numerical result of ASALM and ADM-G for (5.11).

	MRI (128 × 128)				
	SNR_0	SNR	It.	Time(s)	
		ASALM & ADM-G	ASALM & ADM-G	ASALM	ADM-G
$s = 2$	−12.45	−22.85	46	4.14	5.06
$s = 4$	−9.84	−16.30	50	5.48	6.56
$s = 8$	−8.50	−11.67	62	8.81	10.25

	montage (256 × 256)				
	SNR_0	SNR	It.	Time(s)	
		ASALM & ADM-G	ASALM & ADM-G	ASALM	ADM-G
$s = 2$	−16.50	−33.36	30	14.97	18.75
$s = 4$	−14.79	−25.82	41	21.38	26.98
$s = 8$	−13.27	−18.36	50	33.36	39.89

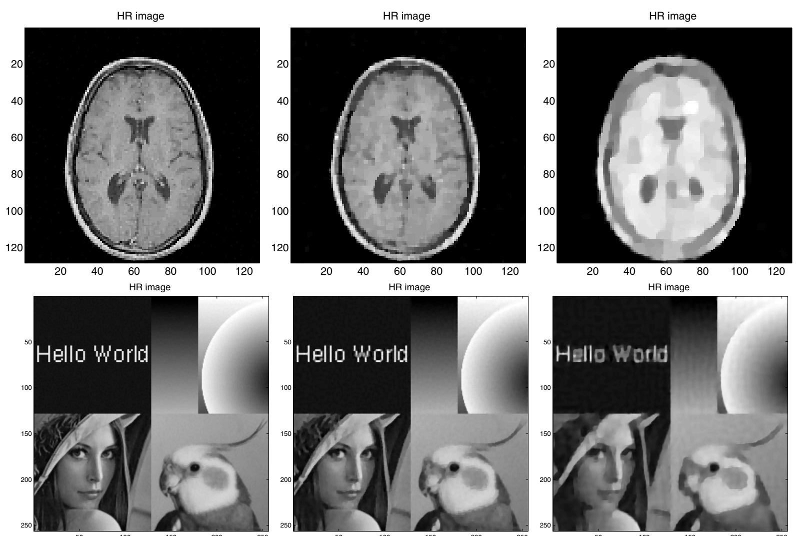

FIG. 5.4. *Top row: Recovered HRIs of MRI by ADM-G and ASALM with* $s = 2, 4, 8$. *Bottom row: Recovered HRIs of montage.png with* $s = 2, 4, 8$.

The SNR in the unit of dB is defined by (see, e.g., [35, Appendix 3])

$$\text{SNR} = 10 \log_{10} \frac{\|\bar{x} - x\|^2}{\|x\|^2},$$

where $\bar{x}$ is the restored image and x is the original.

In Table 5.3, for various values of the down-sampling factor (denoted by "s"), we report the SNR values ("SNR") of the reconstructed images, the number of iterations ("It.") and the computing time in seconds ("Time(s)") when the stopping criterion (5.17) is satisfied. The column of "SNR_0" reflects the SNR values when a simple linear interpolation is performed for the middle frame of the generated LRIs. As we have analyzed, for the application (5.9), ADM-G and ASALM recover the same images by the same number of iterations. Thus, the columns of "SNR" and "It." are the same for these two methods, and the only difference between these two methods is the computing time. In Figure 5.4, we show the reconstructed images by ADM-G and ASALM.

According to Table 5.3 and Figure 5.4, both ADM-G and ASALM are efficient for the TV super-resolution model (5.9), and their effectiveness is exactly the same despite the fact that ADM-G requires moderately more computation in the Gaussian back substitution step.

6. Conclusions. By combining a Douglas–Rachford alternating direction method of multipliers (ADM) with a Gaussian back substitution procedure, this paper develops an efficient method for solving the linearly constrained separable convex minimization problem, whose objective function is separated into m ($m \geq 3$) individual functions with nonoverlapping variables. The efficiency of the new method is shown for solving some concrete applications arising in various disciplines. In the future, we will investigate combinations of various ADM schemes with some substitution procedures and study the convergence rate of such methods.

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

REFERENCES

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

[2] E. Blum and W. Oettli, *Mathematische Optimierung*, Econometrics and Operations Research XX, Springer-Verlag, Berlin, New York, 1975.

[3] N. Bose and K. Boo, *High-resolution image reconstruction with multisensors*, Int. J. Imag. Syst. Tech., 9 (1998), pp. 294–304.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2010), pp. 1–122.

[5] E. J. Candés, X. D. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, J. ACM, 58 (2011), 11.

[6] E. J. Candés and T. Tao, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2009), pp. 2053–2080.

[7] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, *Rank-sparsity incoherence for matrix decomposition*, SIAM J. Optim., 21 (2011), pp. 572–596.

[8] C. H. Chen, B. S. He, and X. M. Yuan, *Matrix completion via an alternating direction method*, IMA J. Numer. Anal., 32 (2012), pp. 227–245.

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[10] J. Chung, E. Haber, and J. Nagy, *Numerical methods for coupled super-resolution*, Inverse Problems, 22 (2006), pp. 1261–1272.

[11] J. Eckstein, *Some saddle-function splitting methods for convex programming*, Optim. Methods Softw., 4 (1994), pp. 75–83.

[12] E. Esser, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, UCLA CAM Report 09-31, University of California, Los Angeles, 2009.

[13] M. Fukushima, *Application of the alternating direction method of multipliers to separable convex programming problems*, Comput. Optim. Appl., 2 (1992), pp. 93–111.

[14] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[15] R. Glowinski and A. Marrocco, *Approximation par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes non linéaires*, RAIRO Anal. Numer., 9 (1975), pp. 41–76.

[16] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Stud. Appl. Math. 9, Philadelphia, 1989.

[17] D. Goldfarb and S. Q. Ma, *Fast multiple splitting algorithms for convex optimization*, SIAM J. Optim., to appear.

[18] D. R. Han, X. M. Yuan, and W. X. Zhang, *An Augmented-Lagrangian-based Parallel Splitting Method for Linearly Constrained Separable Convex Programming with Applications to Image Processing*, manuscript.

[19] P. C. Hansen, J. G. Nag, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, 2006.

[20] B. S. He, L. Z. Liao, D. R. Han, and H. Yang, *A new inexact alternating directions method for monontone variational inequalities*, Math. Program. Ser. A, 92 (2002), pp. 103–118.

[21] B. S. He, M. Tao, M. H. Xu, and X. M. Yuan, *An alternating direction-based contraction method for linearly constrained separable convex programming problems*, Optimization, to appear.

[22] B. S. He, M. Tao, and X. M. Yuan, *A splitting method for separable convex programming*, IMA J. Numer. Anal., submitted.

[23] B. S. He, S. L. Wang, and H. Yang, *A modified variable-penalty alternating directions method for monotone variational inequalities*, J. Comput. Math., 21 (2003), pp. 495–504.

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

[25] B. S. He, M. H. Xu, and X. M. Yuan, *Solving large-scale least squares semidefinite programming by alternating direction methods*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 136–152.

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

[27] M. Kang and S. Chaudhuri, *Super-resolution image reconstruction*, IEEE Signal Process., 20 (2003), pp. 19–20.

[28] S. Kontogiorgis and R. R. Meyer, *A variable-penalty alternating directions method for convex optimization*, Math. Program., 83 (1998), pp. 29–53.

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

[30] Mosek ApS Inc., *The Mosek Optimization Tools*, Version 6, 2009.

[31] A. Nedić and A. Ozdaglar, *Cooperative distributed multi-agent optimization*, in Convex Optimization in Signal Processing and Communications, D. P. Palomar and Y. C. Eldar, eds., Cambridge University Press, Cambridge, UK, 2010, pp. 340–386.

[32] M. K. Ng, P. Weiss, and X. M. Yuan, *Solving constrained total-variation image restoration problems via alternating direction methods*, SIAM J. Sci. Comput., 32 (2010), pp. 2710–2736.

[33] Y. G. Peng, A. Ganesh, J. Wright, W. L. Xu, and Y. Ma, *Robust alignment by sparse and low-rank decomposition for linearly correlated images*, IEEE Trans. Pattern Anal. Mach. Intell., to appear.

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

[35] W. K. Pratt, *Digital Image Processing: PIKS Inside*, 3rd ed., John Wiley & Sons, New York, 2001.

[36] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.

[37] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D., 60 (1992), pp. 259–268.

[38] S. Setzer, G. Steidl, and T. Teuber, *Deblurring Poissonian images by split Bregman techniques*, J. Visual Commun. Image Represent., 21 (2010), pp. 193–199.

[39] J. Sun and S. Zhang, *A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs*, European J. Oper. Res., 207 (2010), pp. 1210–1220.

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

[41] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, *Sparsity and smoothness via the fused lasso*, J. Royal Statist. Soc., 67 (2005), pp. 91–108.

[42] A. N. Tikhonov, *On the solution of ill-posed problems and the method of regularization*, Dokl. Akad. Nauk SSSR, 151 (1963), pp. 501–504 (in Russian).

[43] Z. Wen, D. Goldfarb, and W. Yin, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Math. Program. Comput., 2 (2010), pp. 203–230.

[44] J. Yang and Y. Zhang, *Alternating direction algorithms for ℓ_1-problems in compressive sensing*, SIAM J. Sci. Comput., 33 (2011), pp. 250–278.

[45] C.-H. Ye and X.-M. Yuan, *A descent method for structured monotone variational inequalities*, Optim. Methods Softw., 22 (2007), pp. 329–338.