AIMS Mathematics, 8(9): 21531–21553.
DOI: 10.3934/math.20231097
Received: 01 May 2023
Revised: 16 June 2023
Accepted: 24 June 2023
Published: 06 July 2023

http://www.aimspress.com/journal/Math

Research article

Iterative methods to solve the constrained Sylvester equation

Siting Yu, Jingjing Peng*, Zengao Tang and Zhenyun Peng

College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004, China

* **Correspondence:** Email: jjpeng2012@163.com.

Abstract: In this paper, the multiple constraint least squares solution of the Sylvester equation $AX + XB = C$ is discussed. The necessary and sufficient conditions for the existence of solutions to the considered matrix equation are given. Noting that the alternating direction method of multipliers (ADMM) is a one-step iterative method, a multi-step alternating direction method of multipliers (MSADMM) to solve the considered matrix equation is proposed and some convergence results of the proposed algorithm are proved. Problems that should be studied in the near future are listed. Numerical comparisons between MSADMM, ADMM and ADMM with Anderson acceleration (ACADMM) are included.

Keywords: nonlinear matrix equation; fixed point iteration algorithm; Anderson acceleration algorithm; Thompson distance
Mathematics Subject Classification: 15A24, 65F30

1. Introduction

The Sylvester equation

$$AX + XB = C \tag{1.1}$$

appears frequently in many areas of applied mathematics. We refer readers to the elegant survey by Bhatia and Rosenthal [1] and the references therein for the history of the Sylvester equation and many interesting and important theoretical results. The Sylvester equation is important in a number of applications such as matrix eigenvalue decompositions [2,3], control theory [3–5], model reduction [6–9], physics mathematics to construct exact solutions of nonlinear integrable equations [10], feature problemss of slice semi-regular functions [11] and the numerical solution of the matrix differential Riccati equations [12–14]. There are several numerical algorithms to compute the solution of the Sylvester equation. The standard ones are the Bartels Stewart algorithm [15] and the Hessenberg Schur

method first described by Enright [14], but more often attributed to Golub, Nash and Van Loan [16]. Other computationally efficient approaches for the case that both A and B are stable, i.e., both A and B have all their eigenvalues in the open left half plane, are the sign function method [17], Smith method [18] and ADI iteration methods [19–22]. All these methods are efficient for the small size of the dense matrices A and B.

The recent interest is directed more towards the large and sparse matrices A and B, and C with low rank. For the dense A and B, the approach based on the sign function method is suggested in [23] that exploits the low rank structure of C. This approach is further used in [24] in order to solve the large scale Sylvester equation with sparse A and B, i.e., the matrices A and B can be represented by $O(nlog(n))$ data. Problems for the sensitivity of the solution of the Sylvester equation are also widely studied. There are several books that contain the results for these problems [25–27].

In this paper, we focus our attention on the multiple constrained least squares solution of the Sylvester equation, that is, the following multiple constrained least squares problem:

$$\min_{X^T=X, L\leq X\leq U, \lambda_{min}(X)\geq\varepsilon>0} f(X) = \frac{1}{2}||AX + XB - C||^2 \tag{1.2}$$

where A, B, C, L and U are given $n\times n$ real matrices, X is a $n\times n$ real symmetric matrix which we wish to find, $\lambda_{min}(X)$ represents the smallest eigenvalue of the symmetric matrix X, and ε is a given positive constant. The inequality $X \geq Y$, for any two real matrices, means that $X_{ij} \geq Y_{ij}$, here X_{ij} and Y_{ij} denote the ijth entries of the matrices X and Y, respectively.

Multiple constrained conditions least squares estimations of matrices are widely used in mathematical economics, statistical data analysis, image reconstruction, recommendation problems and so on. They differ from the ordinary least squares problems, and the estimated matrices are usually required to be symmetric positive definite, bounded and, sometimes, to have some special construction patterns. For example, in the dynamic equilibrium model of economy [28], one needs to estimate an aggregate demand function derived from second order analysis of the utility function of individuals. The formulation of this problem is to find the least squares solution of the matrix equation $AX = B$, where A and B are given, the fitting matrix X is a symmetric and bounded matrix, and the smallest eigenvalue is no less than a specified positive number since, in the neighborhood of equilibrium, the approximate of the utility function is a quadratic and strictly concave with Hessian matrix. Other examples discussed in [29, 30] are respectively to find a symmetric positive definite patterned matrix closest to a sample covariance matrix and to find a symmetric and diagonally dominant matrices with positive diagonal matrix closest to a given matrix. Based on the above analysis, we have a strong motivation to study the multiple constrained least squares problem (1.2).

In this paper, we first transform the multiple constrained least squares (1.2) into an equivalent constrained optimization problem. Then, we give the necessary and sufficient conditions for the existence of a solution to the equivalent constrained optimization problem. Noting that the alternating direction method of multipliers (ADMM) is one-step iterative method, we propose a multi-step alternating direction method of multipliers (MSADMM) to the multiple constrained least squares (1.2), and analyze the global convergence of the proposed algorithm. We will give some numerical examples to illustrate the effectiveness of the proposed algorithm to the multiple constrained least squares (1.2) and list some problems that should be studied in the near future. We also give some numerical comparisons between MSADMM, ADMM and ADMM with Anderson acceleration (ACADMM).

Throughout this paper, $R^{m\times n}$, $SR^{n\times n}$ and $SR_0^{n\times n}$ denote the set of $m\times n$ real matrices, $n\times n$ symmetric matrices and $n\times n$ symmetric positive semidefinite matrices, respectively. I_n stands for the $n\times n$ identity matrix. A_+ denotes a matrix with ijth entry equal to $\max\{0, A_{ij}\}$. The inner product in space $R^{m\times n}$ defined as $\langle A, B\rangle = tr(A^TB) = \sum_{ij} A_{ij}B_{ij}$ for all $A, B \in R^{m\times n}$, and the associated norm is Frobenius norm denoted by $\|A\|$. $P_\Omega(X)$ denotes the projection of the matrix X onto the constrained matrix set Ω, that is $P_\Omega(X) = arg\min_{Z\in\Omega}\|Z - X\|$.

2. Preliminaries

In this section, we give an existence theorem for a solution of the multiple constrained least squares problem (1.2) and some theoretical results for the optimization problems which are useful for discussions in the next sections.

Theorem 2.1. *The matrix $\tilde{X}$ is a solution of the multiple constrained least squares problem (1.2) if and only if there exist matrices $\tilde{\Lambda}_1, \tilde{\Lambda}_2$ and $\tilde{\Lambda}_3$ such that the following conditions (2.1)–(2.4) are satisfied.*

$$A^T(A\tilde{X} + \tilde{X}B - C) + (A\tilde{X} + \tilde{X}B - C)B^T - \tilde{\Lambda}_1 + \tilde{\Lambda}_2 - \tilde{\Lambda}_3 = 0, \tag{2.1}$$

$$\langle\tilde{\Lambda}_1, \tilde{X} - L\rangle = 0, \tilde{X} - L \geq 0, \tilde{\Lambda}_1 \geq 0, \tag{2.2}$$

$$\langle\tilde{\Lambda}_2, U - \tilde{X}\rangle = 0, U - \tilde{X} \geq 0, \tilde{\Lambda}_2 \geq 0, \tag{2.3}$$

$$\left\langle\ \tilde{\Lambda}_3 + \tilde{\Lambda}_3^T, \tilde{X} - \varepsilon I_n\ \right\rangle = 0, \tilde{X} - \varepsilon I_n \in SR_0^{n\times n}, \tilde{\Lambda}_3 + \tilde{\Lambda}_3^T \in SR_0^{n\times n}. \tag{2.4}$$

Proof. Obviously, the multiple constrained least squares problem (1.2) can be rewritten as

$$\begin{aligned}&\min_{X\in SR^{n\times n}} F(X) = \frac{1}{2}\|AX + XB - C\|^2\\ &\text{s.t. } X - L \geq 0, U - X \geq 0, X - \varepsilon I_n \in SR_0^{n\times n}.\end{aligned} \tag{2.5}$$

Then, if $\tilde{X}$ is a solution to the constrained optimization problem (2.5), $\tilde{X}$ certainly satisfies KKT conditions of the constrained optimization problem (2.5), and hence of the multiple constrained least squares problem (1.2). That is, there exist matrices $\tilde{\Lambda}_1, \tilde{\Lambda}_2$ and $\tilde{\Lambda}_3$ such that conditions (2.1)–(2.4) are satisfied.

Conversely, assume that there exist matrices $\tilde{\Lambda}_1, \tilde{\Lambda}_2$ and $\tilde{\Lambda}_3$ such that conditions (2.1)–(2.4) are satisfied. Let

$$\bar{F}(X) = F(X) - \langle\tilde{\Lambda}_1, X - L\rangle - \langle\tilde{\Lambda}_2, U - X\rangle - \left\langle\ \tfrac{\tilde{\Lambda}_3+\tilde{\Lambda}_3^T}{2}, X - \varepsilon I_n\ \right\rangle,$$

then, for any matrix $W \in SR^{n\times n}$, we have

$$\begin{aligned}&\bar{F}(\tilde{X} + W)\\ &= \frac{1}{2}\|A(\tilde{X} + W) + (\tilde{X} + W)B - C\|^2\end{aligned}$$

$$
\begin{aligned}
&-\langle\tilde{\Lambda}_1, \tilde{X}+W-L\rangle-\langle\tilde{\Lambda}_2, U-\tilde{X}-W\rangle-\left\langle \tfrac{\tilde{\Lambda}_3+\tilde{\Lambda}_3^T}{2}, \tilde{X}+W-\varepsilon I_n \right\rangle \\
&=\bar{F}(\tilde{X})+\frac{1}{2}\|AW+WB\|^2+\langle AW+WB, A\tilde{X}+\tilde{X}B-C\rangle \\
&\quad-\langle\tilde{\Lambda}_1, W\rangle+\langle\tilde{\Lambda}_2, W\rangle-\left\langle \tfrac{\tilde{\Lambda}_3+\tilde{\Lambda}_3^T}{2}, W \right\rangle \\
&=\bar{F}(\tilde{X})+\frac{1}{2}\|AW+WB\|^2 \\
&\quad+\langle W, A^T(A\tilde{X}+\tilde{X}B-C)+(A\tilde{X}+\tilde{X}B-C)B^T-\tilde{\Lambda}_1+\tilde{\Lambda}_2-\tilde{\Lambda}_3\rangle \\
&=\bar{F}(\tilde{X})+\frac{1}{2}\|AW+WB\|^2 \\
&\geq \bar{F}(\tilde{X}).
\end{aligned}
$$

This implies that $\tilde{X}$ is a global minimizer of the function $\bar{F}(X)$ with $X \in SR^{n\times n}$. Since

$$\langle\tilde{\Lambda}_1, \tilde{X}-L\rangle=0, \langle\tilde{\Lambda}_2, U-\tilde{X}\rangle=0, \left\langle \tilde{\Lambda}_3+\tilde{\Lambda}_3^T, \tilde{X}-\varepsilon I_n \right\rangle=0,$$

and $\bar{F}(X) \geq \bar{F}(\tilde{X})$ holds for all $X \in SR^{n\times n}$, we have

$$F(X) \geq F(\tilde{X})+\langle\tilde{\Lambda}_1, X-L\rangle+\langle\tilde{\Lambda}_2, U-X\rangle+\left\langle \tfrac{\tilde{\Lambda}_3+\tilde{\Lambda}_3^T}{2}, X-\varepsilon I_n \right\rangle.$$

Noting that $\tilde{\Lambda}_1 \geq 0, \tilde{\Lambda}_2 \geq 0$ and $\tilde{\Lambda}_3+\tilde{\Lambda}_3^T \in SR_0^{n\times n}$, then $F(X) \geq F(\tilde{X})$ holds for all X with $X-L \geq 0$, $U-X \geq 0$ and $X-\varepsilon I_n \in SR_0^{n\times n}$. Hence, $\tilde{X}$ is a solution of the constrained optimization problem (2.5), that is, $\tilde{X}$ a solution of the multiple constrained least squares problem (1.2). □

Lemma 2.1. *[31] Assume that $\tilde{x}$ is a solution of the optimization problem*

$$\min f(x) \ \ s.t. \ x \in \Omega,$$

where $f(x)$ is a continuously differentiable function, Ω is a closed convex set, then

$$\langle\nabla f(\tilde{x}), x-\tilde{x}\rangle \geq 0, \ \forall x \in \Omega.$$

Lemma 2.2. *[31] Assume that $(\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ is a solution of the optimization problem*

$$\min \sum_{i=1}^{n} f_i(x_i) \ \ s.t. \ \sum_{i=1}^{n} A_i x_i = b, \ x_i \in \Omega_i, \ i=1,2,...,n \tag{2.6}$$

where $f_i(x_i)$ $(i=1,2,...,n)$ are continuously differentiable functions, Ω_i $(i=1,2,...,n)$ are closed convex sets, then

$$\langle\nabla_{x_i} f(\tilde{x}_i)-A_i^T\tilde{\lambda}, x_i-\tilde{x}_i\rangle \geq 0, \ \forall x_i \in \Omega_i, \ i=1,2,...,n,$$

where $\tilde{\lambda}$ is a solution to the dual problem of (2.6).

Lemma 2.3. *Assume that $\Omega=\{X \in R^{n\times n} : L \leq X \leq U\}$, then, for any matrix $M \in R^{n\times n}$, if $Y=P_\Omega(Y-M)$, we have*

$$\langle(M)_+, Y-L\rangle=0, \ \langle(-M)_+, U-Y\rangle=0.$$

Proof. Let

$$\tilde{Z} = arg \min_{Z\in\Omega} \|Z - (Y - M)\|^2$$

and noting that the optimization problem

$$\min_{Z\in\Omega} \|Z - (Y - M)\|^2$$

is equivalent to the optimization problem

$$\min_{Z\in R^{n\times n}, Z-L\geq 0, U-Z\geq 0} \|Z - (Y - M)\|^2, \tag{2.7}$$

then $\tilde{Z}$ satisfies the KKT conditions for the optimization problem (2.7). That is, there exist matrices $\tilde{\Lambda}_1 \geq 0$ and $\tilde{\Lambda}_2 \geq 0$ such that

$$\tilde{Z} - Y + M - \tilde{\Lambda}_1 + \tilde{\Lambda}_2 = 0, \langle \tilde{\Lambda}_1, \tilde{Z} - L\rangle = 0, \langle \tilde{\Lambda}_2, U - \tilde{Z}\rangle = 0, \tilde{Z} - L \geq 0,\ U - \tilde{Z} \geq 0.$$

Since

$$Y = P_\Omega(Y - M) = arg \min_{Z\in\Omega} \|Z - (Y - M)\|^2,$$

then

$$M - \tilde{\Lambda}_1 + \tilde{\Lambda}_2 = 0,\ \langle \tilde{\Lambda}_1, Y - L\rangle = 0,\ \langle \tilde{\Lambda}_2, U - Y\rangle = 0,\ Y - L \geq 0,\ U - Y \geq 0.$$

So we have from the above conditions that $(\tilde{\Lambda}_1)_{ij}(\tilde{\Lambda}_2)_{ij} = 0$ when $L_{ij} \neq U_{ij}$, and $(\tilde{\Lambda}_1)_{ij}$ and $(\tilde{\Lambda}_2)_{ij}$ can be arbitrarily selected as $(\tilde{\Lambda}_1)_{ij} \geq 0$ and $(\tilde{\Lambda}_2)_{ij} \geq 0$ when $L_{ij} = U_{ij}$. Noting that $M = \tilde{\Lambda}_1 - \tilde{\Lambda}_2$, $\tilde{\Lambda}_1$ and $\tilde{\Lambda}_2$ can be selected as $\tilde{\Lambda}_1 = (M)_+$ and $\tilde{\Lambda}_2 = (-M)_+$. Hence, the results hold. □

Lemma 2.4. *Assume that* $\Omega = \{X \in R^{n\times n} : X^T = X, \lambda_{min}(X) \geq \varepsilon > 0\}$*, then, for any matrix* $M \in R^{n\times n}$*, if* $Y = P_\Omega(Y - M)$*, we have*

$$\langle M + M^T, Y - \varepsilon I_n\rangle = 0,\ M + M^T \in SR_0^{n\times n}, Y - \varepsilon I_n \in SR_0^{n\times n}.$$

Proof. Let

$$\tilde{Z} = arg \min_{Z\in\Omega} \|Z - (Y - M)\|^2$$

and noting that the optimization problem

$$\min_{Z\in\Omega} \|Z - (Y - M)\|^2$$

is equivalent to the optimization problem

$$\min_{Z\in SR^{n\times n}, Z-\varepsilon I_n\in SR_0^{n\times n}} \|Z - (Y - M)\|^2, \tag{2.8}$$

then $\tilde{Z}$ satisfies the KKT conditions for the optimization problem (2.8). That is, there exists a matrix $\tilde{\Lambda} \in SR_0^{n\times n}$ such that

$$\tilde{Z} - Y + M - \tilde{\Lambda} + (\tilde{Z} - Y + M - \tilde{\Lambda})^T = 0, \langle \tilde{\Lambda}, \tilde{Z} - \varepsilon I_n\rangle = 0,\ \tilde{Z} - \varepsilon I_n \in SR_0^{n\times n}.$$

Since

$$Y = P_\Omega(Y - M) = arg \min_{Z\in\Omega} \|Z - (Y - M)\|^2,$$

then

$$M + M^T - 2\tilde{\Lambda} = 0,\ \langle \tilde{\Lambda}, Y - \varepsilon I_n\rangle = 0,\ Y - \varepsilon I_n \in SR_0^{n\times n}.$$

Hence, the results hold. □

3. Accelerated ADMM

In this section we give a multi-step alternating direction method of multipliers (MSADM) tothe multiple constrained least squares problem (1.2). Obviously, the multiple constrained least squares problem (1.2) is equivalent to the following constrained optimization problem

$$\begin{aligned} &\min_X F(X) = \frac{1}{2}\|AX + XB - C\|^2, \\ &\text{s.t. } X - Y = 0, X - Z = 0, \\ &\quad X \in R^{n\times n}, \\ &\quad Y \in \Omega_1 = \{Y \in R^{n\times n} : L \le Y \le U\}, \\ &\quad Z \in \Omega_2 = \{Z \in R^{n\times n} : Z^T = Z, \lambda_{min}(Z) \ge \varepsilon > 0\}. \end{aligned} \tag{3.1}$$

The Lagrange function, augmented Lagrangian function and dual problem to the constrained optimization problem (3.1) are, respectively,

$$L(X, Y, Z, M, N) = F(X) - \langle M, X - Y\rangle - \langle N, X - Z\rangle, \tag{3.2}$$

$$L_\alpha(X, Y, Z, M, N) = F(X) + \frac{\alpha}{2}\|X-Y-M/\alpha\|^2 + \frac{\alpha}{2}\|X-Z-N/\alpha\|^2, \tag{3.3}$$

$$\max_{M,N\in R^{n\times n}} \inf_{X\in R^{n\times n}, Y\in\Omega_1, Z\in\Omega_2} L(X, Y, Z, M, N), \tag{3.4}$$

where M and N are Lagrangian multipliers and α is penalty parameter.

The alternating direction method of multipliers [32, 33] to the constrained optimization problem (3.1) can be described as the following Algorithm 3.1.

Algorithm 3.1. *ADMM to solve problem (3.1).*

Step 1. *Input matrices A, B, C, L and U. Input constant $\varepsilon > 0$, error tolerance $\varepsilon_{out} > 0$ and penalty parameter $\alpha > 0$. Choose initial matrices $Y_0, Z_0, M_0, N_0 \in R^{n\times n}$. Let $k =: 0$;*
Step 2. *Compute*

$$\begin{aligned} &(a)\ X_{k+1} = arg\min_{X\in R^{n\times n}} L_\alpha(X, Y_k, Z_k, M_k, N_k), \\ &(b)\ Y_{k+1} = arg\min_{Y\in\Omega_1} L_\alpha(X_{k+1}, Y, Z_k, M_k, N_k) = P_{\Omega_1}(X_{k+1} - M_k/\alpha), \\ &(c)\ Z_{k+1} = arg\min_{Z\in\Omega_2} L_\alpha(X_{k+1}, Y_{k+1}, Z, M_k, N_k) = P_{\Omega_2}(X_{k+1} - N_k/\alpha), \\ &(d)\ M_{k+1} = M_k - \alpha(X_{k+1} - Y_{k+1}), \\ &(e)\ N_{k+1} = N_k - \alpha(X_{k+1} - Z_{k+1}); \end{aligned} \tag{3.5}$$

Step 3. *If $(\|Y_{k+1} - Y_k\|^2 + \|Z_{k+1} - Z_k\|^2 + \|M_{k+1} - M_k\|^2 + \|N_{k+1} - N_k\|^2)^{1/2} < \varepsilon_{out}$, stop. In this case, X_{k+1} is an approximate solution of problem (3.1);*
Step 4. *Let $k =: k + 1$ and go to step 2.*

Alternating direction method of multipliers (ADMM) has been well studied in the context of the linearly constrained convex optimization. In the last few years, we have witnessed a number of novel applications arising from image processing, compressive sensing and statistics, etc. ADMM is a splitting version of the augmented Lagrange method (ALM) where the ALM subproblem is decomposed into multiple subproblems at each iteration, and thus the variables can be solved separably in alternating order. ADMM, in fact, is one-step iterative method, that is, the current iterates is obtained by the information only from the previous step, and the convergence rate of ADMM is only linear, which was proved in [33]. In this paper we propose a multi-step alternating direction method of multipliers (MSADMM), which is more effective than ADMM, to the constrained optimization problem (3.1). The iterative pattern of MSADMM can be described as the following Algorithm 3.2.

Algorithm 3.2. *MSADMM to solve problem (3.1).*

Step 1. *Input matrices A, B, C, L and U. Input constant $\varepsilon > 0$, error tolerance $\varepsilon_{out} > 0$, penalty parameter $\alpha > 0$ and correction factor $\gamma \in (0, 2)$. Choose initial matrices $Y_0, Z_0, M_0, N_0 \in R^{n\times n}$. Let $k =: 0$;*

Step 2. *ADMM step*

$$(a)\ \tilde{X}_k = arg \min_{X\in R^{n\times n}} L_\alpha(X, Y_k, Z_k, M_k, N_k), \tag{3.6}$$

$$(b)\ \tilde{M}_k = M_k - \alpha(\tilde{X}_k - Y_k), \tag{3.7}$$

$$(c)\ \tilde{N}_k = N_k - \alpha(\tilde{X}_k - Z_k), \tag{3.8}$$

$$(d)\ \tilde{Y}_k = arg \min_{Y\in\Omega_1} L_\alpha(\tilde{X}_k, Y, Z_k, \tilde{M}_k, \tilde{N}_k) = P_{\Omega_1}(\tilde{X}_k - \tilde{M}_k/\alpha), \tag{3.9}$$

$$(e)\ \tilde{Z}_k = arg \min_{Z\in\Omega_2} L_\alpha(\tilde{X}_k, \tilde{Y}_k, Z, \tilde{M}_k, \tilde{N}_k) = P_{\Omega_2}(\tilde{X}_k - \tilde{N}_k/\alpha); \tag{3.10}$$

Step 3. *Correction step*

$$(a)\ Y_{k+1} = Y_k - \gamma(Y_k - \tilde{Y}_k), \tag{3.11}$$

$$(b)\ Z_{k+1} = Z_k - \gamma(Z_k - \tilde{Z}_k), \tag{3.12}$$

$$(c)\ M_{k+1} = M_k - \gamma(M_k - \tilde{M}_k). \tag{3.13}$$

$$(d)\ N_{k+1} = N_k - \gamma(N_k - \tilde{N}_k); \tag{3.14}$$

Step 4. *If $(\|Y_{k+1} - Y_k\|^2 + \|Z_{k+1} - Z_k\|^2 + \|M_{k+1} - M_k\|^2 + \|N_{k+1} - N_k\|^2)^{1/2} < \varepsilon_{out}$, stop. In this case, $\tilde{X}_k$ is an approximate solution of problem (3.1);*

Step 5. *Let $k =: k + 1$ and go to step 2.*

Compared to ADMM, MSADMM yields the new iterate in the order $X \to M \to N \to Y \to Z$ with the difference in the order of $X \to Y \to Z \to M \to N$. Despite this difference, MSADMM and ADMM are equally effective to exploit the separable structure of (3.1) and equally easy to implement. In fact, the resulting subproblems of these two methods are of the same degree of decomposition and they are of the same difficulty. We shall verify by numerical experiments that these two methods are also equally competitive in numerical senses, and that, if we choose the correction factor γ suitably, MSADMM is more efficient than ADMM.

4. Convergence analysis

In this section, we prove the global convergence and the $O(1/t)$ convergence rate for the proposed Algorithm 3.2. We start the proof with some lemmas which are useful for the analysis of coming theorems.

To simplify our analysis, we use the following notations throughout this section.

$$\Omega = R^{n\times n} \times \Omega_1 \times \Omega_2 \times R^{n\times n} \times R^{n\times n}; \mathcal{V} = \begin{pmatrix} Y \\ Z \\ M \\ N \end{pmatrix}; \mathcal{W} = \begin{pmatrix} X \\ \mathcal{V} \end{pmatrix};$$

$$G(\mathcal{W}) = \begin{pmatrix} \nabla F(X) - M - N \\ M \\ N \\ X - Y \\ X - Z \end{pmatrix};$$

$$Q = \begin{pmatrix} \alpha I_n & 0 & I_n & 0 \\ 0 & \alpha I_n & 0 & I_n \\ I_n & 0 & \frac{1}{\alpha} I_n & 0 \\ 0 & I_n & 0 & \frac{1}{\alpha} I_n \end{pmatrix}.$$

Lemma 4.1. *Assume that (X^*, Y^*, Z^*) is a solution of problem (3.1), (M^*, N^*) is a solution of the dual problem (3.4) to the constrained optimization problem (3.1), and that the sequences $\{\mathcal{V}_k\}$ and $\{\tilde{\mathcal{W}}_k\}$ are generated by Algorithm 3.2, then we have*

$$\langle \mathcal{V}_k - \mathcal{V}^*, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle \geq \langle \mathcal{V}_k - \tilde{\mathcal{V}}_k, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle. \tag{4.1}$$

Proof. By (3.6)–(3.10) and Lemma 2.1, we have, for any $(X, Y, Z, M, N) \in \Omega$, that

$$\left\langle \begin{pmatrix} X - \tilde{X}_k \\ Y - \tilde{Y}_k \\ Z - \tilde{Z}_k \\ M - \tilde{M}_k \\ N - \tilde{N}_k \end{pmatrix}, \begin{pmatrix} \nabla F(\tilde{X}_k) - \tilde{M}_k - \tilde{N}_k \\ \tilde{M}_k \\ \tilde{N}_k \\ \tilde{X}_k - \tilde{Y}_k \\ \tilde{X}_k - \tilde{Z}_k \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ \alpha I_n & 0 & I_n & 0 \\ 0 & \alpha I_n & 0 & I_n \\ I_n & 0 & \frac{1}{\alpha} I_n & 0 \\ 0 & I_n & 0 & \frac{1}{\alpha} I_n \end{pmatrix} \begin{pmatrix} \tilde{Y}_k - Y_k \\ \tilde{Z}_k - Z_k \\ \tilde{M}_k - M_k \\ \tilde{N}_k - N_k \end{pmatrix} \right\rangle \geq 0, \tag{4.2}$$

or compactly,

$$\langle \mathcal{W} - \tilde{\mathcal{W}}_k, G(\tilde{\mathcal{W}}_k) + \begin{pmatrix} 0 \\ Q \end{pmatrix} (\tilde{\mathcal{V}}_k - \mathcal{V}_k)\rangle \geq 0. \tag{4.3}$$

Choosing $\mathcal{W}$ as $\mathcal{W}^* = \begin{pmatrix} X^* \\ \mathcal{V}^* \end{pmatrix}$, then (4.3) can be rewritten as

$$\langle \tilde{\mathcal{V}}_k - \mathcal{V}^*, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle \geq \langle \tilde{\mathcal{W}}_k - \mathcal{W}^*, G(\tilde{\mathcal{W}}_k)\rangle.$$

Noting that the monotonicity of the gradients of the convex functions, we have by Lemma 2.2 that

$$\langle \tilde{\mathcal{W}}_k - \mathcal{W}^*, G(\tilde{\mathcal{W}}_k)\rangle \geq \langle \tilde{\mathcal{W}}_k - \mathcal{W}^*, G(\mathcal{W}^*)\rangle \geq 0.$$

Therefore, the above two inequalities imply that

$$\langle \tilde{\mathcal{V}}_k - \mathcal{V}^*, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle \geq 0,$$

from which the assertion (4.1) is immediately derived. □

Noting that the matrix Q is a symmetric and positive semi-definite matrix, we use, for convenience, the notation

$$\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q := \sqrt{\langle \mathcal{V}_k - \tilde{\mathcal{V}}_k, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle}.$$

Then, the assertion (4.1) can be rewritten as

$$\langle \mathcal{V}_k - \mathcal{V}^*, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle \geq \|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2. \tag{4.4}$$

Lemma 4.2. *Assume that (X^*, Y^*, Z^*) is a solution of the constrained optimization problem (3.1), (M^*, N^*) is a solution of the dual problem (3.4) to the constrained optimization problem (3.1), and that the sequences $\{\mathcal{V}_k\}, \{\tilde{\mathcal{V}}_k\}$ are generated by Algorithm 3.2. Then, we have*

$$\|\mathcal{V}_{k+1} - \mathcal{V}^*\|_Q^2 \leq \|\mathcal{V}_k - \mathcal{V}^*\|_Q^2 - \gamma(2-\gamma)\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2. \tag{4.5}$$

Proof. By elementary manipulation, we obtain

$$\begin{aligned}
\|\mathcal{V}_{k+1} - \mathcal{V}^*\|_Q^2 &= \|(\mathcal{V}_k - \mathcal{V}^*) - \gamma(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\|_Q^2 \\
&= \|\mathcal{V}_k - \mathcal{V}^*\|_Q^2 - 2\gamma\langle \mathcal{V}_k - \mathcal{V}^*, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle + \gamma^2\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2 \\
&\leq \|\mathcal{V}_k - \mathcal{V}^*\|_Q^2 - 2\gamma\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2 + \gamma^2\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2 \\
&= \|\mathcal{V}_k - \mathcal{V}^*\|_Q^2 - \gamma(2-\gamma)\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2,
\end{aligned}$$

where the inequality follows from (4.1) and (4.4). □

Lemma 4.3. *The sequences $\{\mathcal{V}_k\}$ and $\{\tilde{\mathcal{W}}_k\}$ generated by Algorithm 3.2 satisfy*

$$\langle \mathcal{W} - \tilde{\mathcal{W}}_k, G(\tilde{\mathcal{W}}_k)\rangle + \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_k\|_Q^2 - \|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2) \geq (1 - \frac{\gamma}{2})\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2 \tag{4.6}$$

for any $(X, Y, Z, M, N) \in \Omega$.

Proof. By (4.2) or its compact form (4.3), we have, for any $(X, Y, Z, M, N) \in \Omega$, that

$$\langle \mathcal{W} - \tilde{\mathcal{W}}_k, G(\tilde{\mathcal{W}}_k)\rangle \geq -\langle \mathcal{W} - \tilde{\mathcal{W}}_k, \begin{pmatrix} 0 \\ Q \end{pmatrix}(\tilde{\mathcal{V}}_k - \mathcal{V}_k)\rangle = \langle \mathcal{V} - \tilde{\mathcal{V}}_k, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle. \tag{4.7}$$

Thus, it suffices to show that

$$\langle \mathcal{V} - \tilde{\mathcal{V}}_k, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle + \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_k\|_Q^2 - \|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2) \geq (1 - \frac{\gamma}{2})\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2. \tag{4.8}$$

By using the formula $2\langle a, Qb\rangle = \|a\|_Q^2 + \|b\|_Q^2 - \|a-b\|_Q^2$, we derive that

$$\langle \mathcal{V} - \mathcal{V}_{k+1}, Q(\mathcal{V}_k - \mathcal{V}_{k+1})\rangle = \frac{1}{2}\|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2 + \frac{1}{2}\|\mathcal{V}_k - \mathcal{V}_{k+1}\|_Q^2 - \frac{1}{2}\|\mathcal{V} - \mathcal{V}_k\|_Q^2. \tag{4.9}$$

Moreover, since (3.11)–(3.14) can be rewritten as $(\mathcal{V}_k - \mathcal{V}_{k+1}) = \gamma(\mathcal{V}_k - \tilde{\mathcal{V}}_k)$, we have

$$\langle \mathcal{V} - \mathcal{V}_{k+1}, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle = \frac{1}{\gamma}\langle \mathcal{V} - \mathcal{V}_{k+1}, Q(\mathcal{V}_k - \mathcal{V}_{k+1})\rangle. \tag{4.10}$$

Combining (4.9) and (4.10), we obtain

$$\langle \mathcal{V} - \mathcal{V}_{k+1}, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle = \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2 - \|\mathcal{V} - \mathcal{V}_k\|_Q^2) + \frac{1}{2\gamma}\|\mathcal{V}_k - \mathcal{V}_{k+1}\|_Q^2. \tag{4.11}$$

On the other hand, we have by using (3.11)–(3.14) that

$$\langle \mathcal{V}_{k+1} - \tilde{\mathcal{V}}_k, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle = (1-\gamma)\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2. \tag{4.12}$$

By adding (4.11) and (4.12), and again using the fact that $(\mathcal{V}_k - \mathcal{V}_{k+1}) = \gamma(\mathcal{V}_k - \tilde{\mathcal{V}}_k)$, we obtain that

$$\begin{aligned}
&\langle \mathcal{V} - \tilde{\mathcal{V}}_k, Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k)\rangle \\
&= \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2 - \|\mathcal{V} - \mathcal{V}_k\|_Q^2) + \frac{1}{2\gamma}\|\mathcal{V}_k - \mathcal{V}_{k+1}\|_Q^2 + (1-\gamma)\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2 \\
&= \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2 - \|\mathcal{V} - \mathcal{V}_k\|_Q^2) + (1 - \frac{\gamma}{2})\|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q^2
\end{aligned}$$

which is equivalent to (4.8). Hence, the lemma is proved. □

Theorem 4.1. *The sequences $\{\mathcal{V}_k\}$ and $\{\tilde{\mathcal{W}}_k\}$ generated by Algorithm 3.2 are bounded, and furthermore, any accumulation point $\tilde{X}$ of the sequence $\{\tilde{X}_k\}$ is a solution of problem (1.2).*

Proof. The inequality (4.5) with the restriction $\gamma \in (0, 2)$ implies that

(i) $\lim_{k\to\infty} \|\mathcal{V}_k - \tilde{\mathcal{V}}_k\|_Q = 0$;

(ii) $\|\mathcal{V}_k - \mathcal{V}^*\|_Q$ is bounded upper.

Recall that the matrix Q is symmetric and positive semi-definite. Thus, we have by the assertion (i) that $Q(\mathcal{V}_k - \tilde{\mathcal{V}}_k) = 0$ $(k \to \infty)$ which, together with (3.7) and (3.8), imply that $\tilde{Y}_k = \tilde{X}_k = \tilde{Z}_k$ $(k \to \infty)$. The assertion (ii) implies that the sequences $\{Y_k\}$, $\{Z_k\}$, $\{M_k\}$ and $\{N_k\}$ are bounded. Equations (3.11)–(3.14) hold, and the sequences $\{Y_k\}$, $\{Z_k\}$, $\{M_k\}$ and $\{N_k\}$ are bounded imply the sequence $\{\tilde{Y}_k\}$, $\{\tilde{Z}_k\}$, $\{\tilde{M}_k\}$ and $\{\tilde{N}_k\}$ are also bounded. Hence, by the clustering theorem and together with (3.11)–(3.14), there exist subsequences $\{\tilde{X}_k\}_{\mathcal{K}}$, $\{\tilde{Y}_k\}_{\mathcal{K}}$, $\{\tilde{Z}_k\}_{\mathcal{K}}$, $\{\tilde{M}_k\}_{\mathcal{K}}$, $\{\tilde{N}_k\}_{\mathcal{K}}$, $\{Y_k\}_{\mathcal{K}}$, $\{Z_k\}_{\mathcal{K}}$, $\{M_k\}_{\mathcal{K}}$ and $\{N_k\}_{\mathcal{K}}$ such that

$$\lim_{k\to\infty,k\in\mathcal{K}} \tilde{X}_k = \tilde{X}, \lim_{k\to\infty,k\in\mathcal{K}} \tilde{Y}_k = \lim_{k\to\infty,k\in\mathcal{K}} Y_k = \tilde{Y}, \lim_{k\to\infty,k\in\mathcal{K}} \tilde{Z}_k = \lim_{k\to\infty,k\in\mathcal{K}} Z_k = \tilde{Z},$$

$$\lim_{k\to\infty,k\in\mathcal{K}} \tilde{M}_k = \lim_{k\to\infty,k\in\mathcal{K}} M_k = \tilde{M}, \lim_{k\to\infty,k\in\mathcal{K}} \tilde{N}_k = \lim_{k\to\infty,k\in\mathcal{K}} N_k = \tilde{N}.$$

Furthermore, we have by (3.7) and (3.8) that

$$\tilde{X} = \tilde{Y} = \tilde{Z}. \tag{4.13}$$

By (3.6)–(3.8), we have

$$A^T(A\tilde{X}_k + \tilde{X}_kB - C) + (A\tilde{X}_k + \tilde{X}_kB - C)B^T - \tilde{M}_k - \tilde{N}_k = 0.$$

So we have

$$A^T(A\tilde{X} + \tilde{X}B - C) + (A\tilde{X} + \tilde{X}B - C)B^T - \tilde{M} - \tilde{N} = 0. \tag{4.14}$$

Let $k \to \infty, k \in \mathcal{K}$, we have by (3.9), (3.10) and (4.13) that

$$\tilde{X} = P_{\Omega_1}(\tilde{X} - \tilde{M}/\alpha), \tilde{X} = P_{\Omega_2}(\tilde{X} - \tilde{N}/\alpha). \tag{4.15}$$

Noting that $\alpha > 0$, we have by (4.15), Lemma 2.3 and Lemma 2.4 that

$$\langle \tilde{M}_+, \tilde{X} - L\rangle = 0, \tilde{X} - L \geq 0, \tag{4.16}$$

$$\langle (-\tilde{M})_+, U - \tilde{X}\rangle = 0, U - \tilde{X} \geq 0, \tag{4.17}$$

and

$$\langle \tilde{N} + \tilde{N}^T, \tilde{X} - \varepsilon I\rangle = 0, \tilde{N} + \tilde{N}^T \in SR_0^{n\times n}, \tilde{X} - \varepsilon I \in SR_0^{n\times n}. \tag{4.18}$$

Let

$$\tilde{\Lambda}_1 = (\tilde{M})_+, \tilde{\Lambda}_2 = (-\tilde{M})_+, \tilde{\Lambda}_3 = \tilde{N},$$

we have by (4.14) and (4.16)–(4.18) that

$$\begin{cases} A^T(A\tilde{X} + \tilde{X}B - C) + (A\tilde{X} + \tilde{X}B - C)B^T - \tilde{\Lambda}_1 + \tilde{\Lambda}_2 - \tilde{\Lambda}_3 = 0 \\ \langle \tilde{\Lambda}_1, \tilde{X} - L\rangle = 0, \tilde{X} - L \geq 0, \tilde{\Lambda}_1 \geq 0, \\ \langle \tilde{\Lambda}_2, U - \tilde{X}\rangle = 0, U - \tilde{X} \geq 0, \tilde{\Lambda}_2 \geq 0, \\ \langle \tilde{\Lambda}_3 + \tilde{\Lambda}_3^T, \tilde{X} - \varepsilon I\rangle = 0, \tilde{X} - \varepsilon I \in SR_0^{n\times n}, \tilde{\Lambda}_3 + \tilde{\Lambda}_3^T \in SR_0^{n\times n}. \end{cases} \tag{4.19}$$

Hence, we have by Theorem 2.1 that $\tilde{X}$ is a solution of problem (1.2). □

Theorem 4.2. *Let the sequences $\{\tilde{\mathcal{W}}_k\}$ be generated by Algorithm 3.2. For an integer $t > 0$, let*

$$\widetilde{\mathcal{W}}_t = \frac{1}{t+1}\sum_{k=0}^{t}\tilde{\mathcal{W}}_k, \tag{4.20}$$

then $\frac{1}{t+1}\sum_{k=0}^{t}(\tilde{X}_k, \tilde{Y}_k, \tilde{Z}_k, \tilde{M}_k, \tilde{N}_k) \in \Omega$ and the inequality

$$\langle \widetilde{\mathcal{W}}_t - \mathcal{W}, G(\mathcal{W})\rangle \leq \frac{1}{2\gamma(t+1)}\|\mathcal{V} - \mathcal{V}_0\|_Q^2 \tag{4.21}$$

holds for any $(X, Y, Z, M, N) \in \Omega$.

Proof. First, for any integer $t > 0$, we have $(\tilde{X}_k, \tilde{Y}_k, \tilde{Z}_k, \tilde{M}_k, \tilde{N}_k) \in \Omega$ for $k = 0, 1, 2, \cdots, t$. Since $\frac{1}{t+1}\sum_{k=0}^{t}\tilde{\mathcal{W}}_k$ can be viewed as a convex combination of $\tilde{\mathcal{W}}_k's$, we obtain

$$\frac{1}{t+1}\sum_{k=0}^{t}(\tilde{X}_k, \tilde{Y}_k, \tilde{Z}_k, \tilde{M}_k, \tilde{N}_k) \in \Omega.$$

Second, since $\gamma \in (0, 2)$, it follows from Lemma 4.3 that

$$\langle \mathcal{W} - \tilde{\mathcal{W}}_k, G(\tilde{\mathcal{W}}_k)\rangle + \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_k\|_Q^2 - \|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2) \geq 0, \forall(\tilde{X}_k, \tilde{Y}_k, \tilde{Z}_k, \tilde{M}_k, \tilde{N}_k) \in \Omega. \tag{4.22}$$

By combining the monotonicity of $G(\mathcal{W})$ with the inequality (4.22), we have

$$\langle \mathcal{W} - \tilde{\mathcal{W}}_k, G(\mathcal{W})\rangle + \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_k\|_Q^2 - \|\mathcal{V} - \mathcal{V}_{k+1}\|_Q^2) \geq 0, \forall(X, Y, Z, M, N) \in \Omega.$$

Summing the above inequality over $k = 0, 1, \cdots, t$, we derive that

$$\langle (t+1)\mathcal{W} - \sum_{k=0}^{t}\tilde{\mathcal{W}}_k, G(\mathcal{W})\rangle + \frac{1}{2\gamma}(\|\mathcal{V} - \mathcal{V}_0\|_Q^2 - \|\mathcal{V} - \mathcal{V}_{t+1}\|_Q^2) \geq 0, \forall(X, Y, Z, M, N) \in \Omega.$$

By dropping the minus term, we have

$$\langle (t+1)\mathcal{W} - \sum_{k=0}^{t}\tilde{\mathcal{W}}_k, G(\mathcal{W})\rangle + \frac{1}{2\gamma}\|\mathcal{V} - \mathcal{V}_0\|_Q^2 \geq 0, \forall(X, Y, Z, M, N) \in \Omega.$$

which is equivalent to

$$\langle \frac{1}{t+1}\sum_{k=0}^{t}\tilde{\mathcal{W}}_k - \mathcal{W}, G(\mathcal{W})\rangle \leq \frac{1}{2\gamma(t+1)}\|\mathcal{V} - \mathcal{V}_0\|_Q^2, \forall(X, Y, Z, M, N) \in \Omega.$$

The proof is completed. □

Noting that problem (3.1) is equivalent to find $(X^*, Y^*, Z^*, M^*, N^*) \in \Omega$ such that the following inequality

$$\langle \mathcal{W} - \mathcal{W}^*, G(\mathcal{W}^*)\rangle \geq 0 \tag{4.23}$$

holds for any $(X, Y, Z, M, N) \in \Omega$. Theorem 4.2 means that, for any initial matrices $Y_0, Z_0, M_0, N_0 \in R^{n\times n}$, the point $\widetilde{\mathcal{W}}_t$ defined in (4.20) satisfies

$$\langle \widetilde{\mathcal{W}}_t - \mathcal{W}^*, G(\mathcal{W}^*)\rangle \leq \frac{\|\mathcal{V}^* - \mathcal{V}_0\|_Q^2}{2\gamma(t+1)},$$

which means the point $\widetilde{\mathcal{W}}_t$ is an approximate solution of (4.23) with the accuracy $O(1/t)$. That is, the convergence rate $O(1/t)$ of the Algorithm 3.2 is established in an ergodic sense.

5. Numerical experiments

In this section, we present some numerical examples to illustrate the convergence of MSADMM to the constrained least squares problem (1.2). All tests were performed by Matlab 7 with 64-bit Windows 7 operating system. In all tests, the constant $\varepsilon = 0.1$, matrices L with all elements are -1 and U with all elements are 3. The matrices A, B and C are randomly generated, i.e., generated in Matlab style as $A = randn(n, n)$, $B = randn(n, n)$, $C = randn(n, n)$. In all algorithms, the initial matrices are chosen as the null matrices. The maximum number of inner iterations and out iterations are restricted to 5000. The error tolerance $\varepsilon_{out} = \varepsilon_{in} = 10^{-9}$ in Algorithms 3.1 and 3.2. The computational methods of the projection $P_{\Omega_i}(X)$ $(i = 1, 2)$ are as follows [38].

$$P_{\Omega_1}(X) = \begin{cases} X_{ij}, \text{ if } L_{ij} \le X_{ij} \le U_{ij} \\ U_{ij}, \text{ if } X_{ij} > U_{ij} \\ L_{ij}, \text{ if } X_{ij} < L_{ij} \end{cases}, \; P_{\Omega_2}(X) = Wdiag(d_1, d_2, \cdots, d_n)W^T$$

where

$$d_i = \begin{cases} \lambda_i(\frac{X+X^T}{2}), \text{ if } \lambda_i(\frac{X+X^T}{2}) \ge \varepsilon \\ \varepsilon, \text{ if } \lambda_i(\frac{X+X^T}{2}) < \varepsilon \end{cases}$$

and W is such that $\frac{X+X^T}{2} = W\Delta W^T$ is spectral decomposition, i.e., $W^TW = I$ and $\Delta = diag(\lambda_1(\frac{X+X^T}{2}), \lambda_2(\frac{X+X^T}{2}), \cdots, \lambda_n(\frac{X+X^T}{2}))$. We use LSQR algorithm described in [34] with necessary modifications to solve the subproblems (3.5) in Algorithm 3.1, (3.6) in Algorithm 3.2 and (6.1) in Algorithm 6.2.

The LSQR algorithm is an effective method to solve consistent linear matrix equation or least square problem of inconsistent linear matrix equation. Using this iterative algorithm, for any initial matrix, a solution can be obtained within finite iteration steps if exact arithmetic is used. In addition, using this iterative algorithm, a solution with minimum Frobenius norm can be obtained by choosing a special kind of initial matrix, and a solution which is nearest to given matrix in Frobenius norm can be obtained by first finding minimum Frobenius norm solution of a new consistent matrix equation. The LSQR algorithm to solve the subproblems (3.5), (3.6) and (6.1) can be described as follows:

Algorithm 5.1. *LSQR algorithm to solve subproblems (3.5) and (3.6).*

Step 1. *Input matrices A, B, C, Y_k, Z_k, M_k and N_k, penalty parameter $\alpha > 0$ and error tolerance ε_{in}. Compute*

$$\eta_1 = \left(\|C\|^2 + \left\| \sqrt{\alpha}Y_k + \frac{M_k}{\sqrt{\alpha}} \right\|^2 + \left\| \sqrt{\alpha}Z_k + \frac{N_k}{\sqrt{\alpha}} \right\|^2 \right)^{1/2},$$

$$U_1^{(1)} = \frac{C}{\eta_1}, \; U_1^{(2)} = \frac{\sqrt{\alpha}Y_k + \frac{M_k}{\sqrt{\alpha}}}{\eta_1}, \; U_1^{(3)} = \frac{\sqrt{\alpha}Z_k + \frac{N_k}{\sqrt{\alpha}}}{\eta_1},$$

$$\xi_1 = \|A^TU_1^{(1)} + U_1^{(1)}B^T + \sqrt{\alpha}U_1^{(2)} + \sqrt{\alpha}U_1^{(3)}\|,$$

$$\Gamma_1 = \frac{A^TU_1^{(1)} + U_1^{(1)}B^T + \sqrt{\alpha}U_1^{(2)} + \sqrt{\alpha}U_1^{(3)}}{\xi_1},$$

$\Phi_1 = \Gamma_1,\ \bar{\phi} = \eta_1,\ \bar{\rho}_1 = \xi_1.$

Let $i =: 1$;

Step 2. *Compute*

$$\eta_{i+1} = (\|A\Gamma_i + \Gamma_i B - \xi_i U_i^{(1)}\|^2 + \|\sqrt{\alpha}\Gamma_i - \xi_i U_i^{(2)}\|^2 + \|\sqrt{\alpha}\Gamma_i - \xi_i U_i^{(3)}\|^2)^{1/2},$$

$$U_{i+1}^{(1)} = \frac{A\Gamma_i + \Gamma_i B - \xi_i U_i^{(1)}}{\eta_{i+1}},\ U_{i+1}^{(2)} = \frac{\sqrt{\alpha}\Gamma_i - \xi_i U_i^{(2)}}{\eta_{i+1}},\ U_{i+1}^{(3)} = \frac{\sqrt{\alpha}\Gamma_i - \xi_i U_i^{(3)}}{\eta_{i+1}},$$

$$\xi_{i+1} = \|A^T U_{i+1}^{(1)} + U_{i+1}^{(1)} B^T + \sqrt{\alpha}U_{i+1}^{(2)} + \sqrt{\alpha}U_{i+1}^{(3)} - \eta_{i+1}\Gamma_i\|,$$

$$\Gamma_{i+1} = \frac{A^T U_{i+1}^{(1)} + U_{i+1}^{(1)} B^T + \sqrt{\alpha}U_{i+1}^{(2)} + \sqrt{\alpha}U_{i+1}^{(3)} - \eta_{i+1}\Gamma_i}{\xi_{i+1}},$$

$$\rho_i = (\bar{\rho}_i^2 + \eta_{i+1}^2)^{1/2},\ c_i = \frac{\bar{\rho}_i}{\rho_i},\ s_i = \frac{\eta_{i+1}}{\rho_i},\ \theta_{i+1} = s_i\xi_{i+1},\ \bar{\rho}_{i+1} = -c_i\xi_{i+1},$$

$$\phi_i = c_i\bar{\phi}_i,\ \bar{\phi}_{i+1} = s_i\bar{\phi}_i,$$

$$X_{i+1} = X_i + \frac{\phi_i}{\rho_i}\Phi_i,\ \Phi_{i+1} = \Gamma_{i+1} - \frac{\theta_{i+1}}{\rho_i}\Phi_i;$$

Step 3. *If* $\|X_{i+1} - X_i\| < \varepsilon_{in}$*, terminate the execution of the algorithm. (In this case,* X_i *is a solution of problem (3.5) or (3.6));*

Step 4. *Let* $i =: i + 1$ *and go to step 2.*

Table 1 reports the average computing time (CPU) of 10 tests of Algorithm 3.1 (ADMM) and Algorithm 3.2 (MSADMM) with penalty parameter $\alpha = n$. Figures 1–4 report the computing time of ADMM with the same size of problem and different penalty parameters α. Figures 5–8 report the computing time of MSADMM with the same size of problem and different correction factor γ. Figure 9 reports the computing time curve of Algorithm 3.1 (ADMM) and Algorithm 3.2 (MSADMM) with different matrix size.

Table 1. Numerical comparisons between MSADMM and ADMM.

α=n	ADMM	MSADMM(γ=0.8)	MSADMM(γ=1.0)	MSADMM(γ=1.5)
20	0.0846	0.1254	0.0865	0.0538
40	0.3935	0.4883	0.3836	0.2676
80	1.3370	1.6930	1.3726	0.8965
100	2.4766	3.1514	2.5015	1.6488
150	5.8780	7.3482	5.8742	3.9154
200	11.6398	14.6023	11.6162	7.6576
300	35.1929	41.4151	32.9488	21.4898
400	83.7386	108.9807	86.0472	56.8144
500	147.5759	183.4758	144.2038	92.3137
600	242.0408	302.6395	241.6225	157.8042

Based on the tests reported in Table 1, Figures 1–9 and many other performed unreported tests which show similar patterns, we have the following results:

Remark 5.1. *The convergence speed of ADMM is directly related to the penalty parameter α. In general, the penalty parameter α in this paper can be chosen as $\alpha \approx n$ (see Figures 1–4). However, how to select the best penalty parameter α is an important problem should be studied future time.*

Remark 5.2. *The convergence speed of MSADMM is direct relation to the penalty parameter α and the correction factor γ. The selection of the penalty parameter α is similar to ADMM since MSADMM is a direct extension of ADMM. For the correction factor γ, as showed in Table 1 and Figures 5–8, aggressive values such as $\gamma \approx 1.5$ are often preferred. However, how to select the best correction factor γ is also an important problem should be studied future time.*

Remark 5.3. *As showed in Table 1 and Figure 9, MSADMM, with the correction factor $\gamma \approx 1.5$ and the penalty parameter α be chosen as the same as ADMM, is more effective than ADMM.*

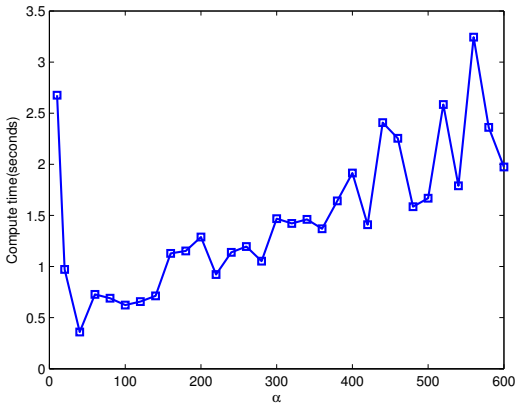

Figure 1. Computing times (seconds) vs. the values of α for n=40.

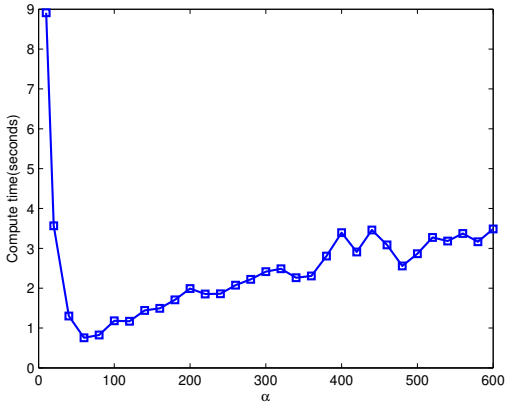

Figure 2. Computing times (seconds) vs. the values of α for n=60.

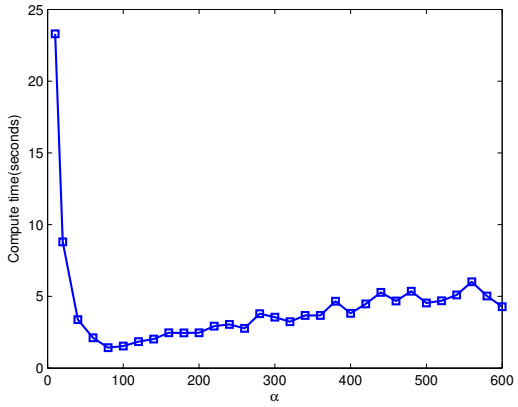

Figure 3. Computing times (seconds) vs. the values of α for n=80.

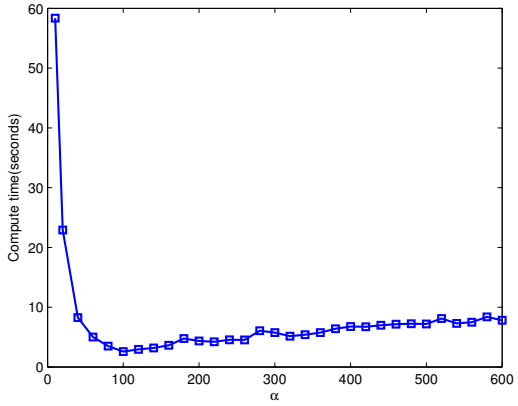

Figure 4. Computing times (seconds) vs. the values of α for n=100.

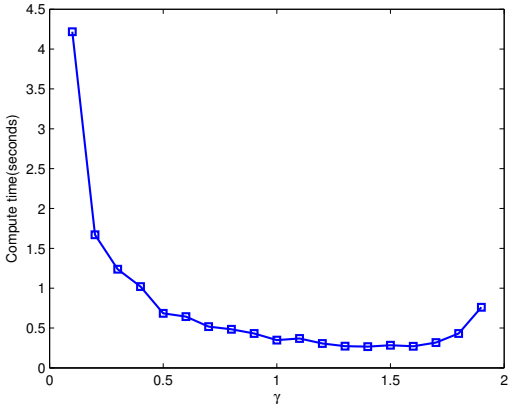

Figure 5. Computing times (seconds) vs. the values of γ for α=n=40.

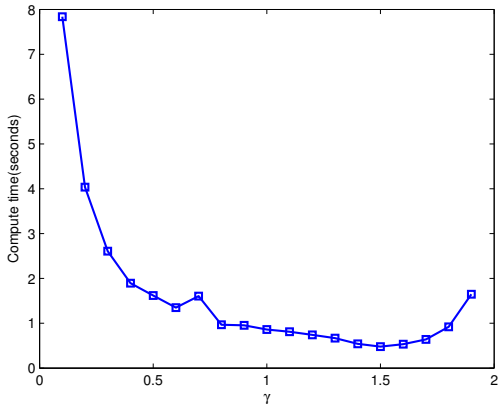

Figure 6. Computing times (seconds) vs. the values of γ for α=n=60.

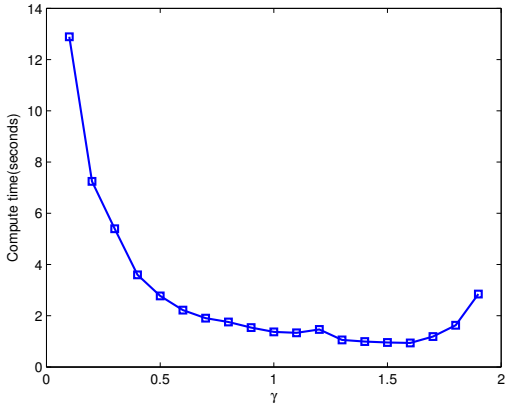

Figure 7. Computing times (seconds) vs. the values of γ for α=n=80.

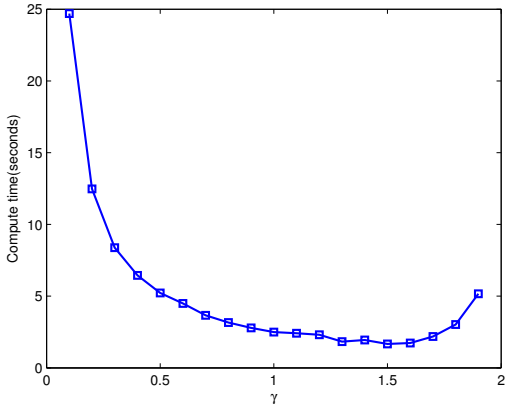

Figure 8. Computing times (seconds) vs. the values of γ for α=n=100.

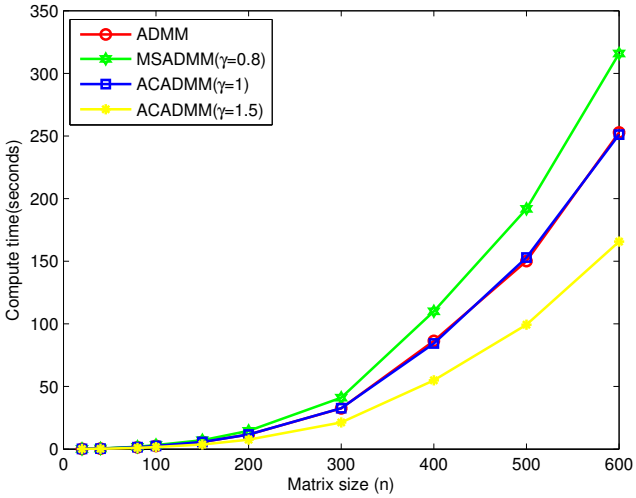

Figure 9. Numerical comparisons between ADMM and MSADMM.

6. Algorithm 3.1 with Anderson acceleration

Anderson acceleration, or Anderson mixing, was initially developed in 1965 by Donald Anderson [35] as an iterative procedure to solve some nonlinear integral equations arising in physics. It turns out that the Anderson acceleration is very efficient to solve other types of nonlinear equations as well, see [36–38], and the literature cited therein. When Anderson acceleration is applied to the equation $f(x) = g(x) - x = 0$, the iterative pattern can be described as the following Algorithm 6.1.

Algorithm 6.1. *Anderson accelerated method to solve the equation* $f(x) = 0$.

Given $x_0 \in R^n$ *and an integer* $m \geq 1$ *this algorithm produces a sequence* x_k *of iterates intended to converge to a fixed point of the function* $g : R^n \to R^n$

Step 1. *Compute* $x_1 = g(x_0)$*;*

Step 2. *For* $k = 1, 2, \cdots$ *until convergence;*

Step 3. *Let* $m_k = \min(m, k)$*;*

Step 4. *Compute* $\lambda_k = (\lambda_1, \lambda_2, \cdots, \lambda_{m_k})^T \in \mathbb{R}^{m_k}$ *that solves*

$$\min_{\lambda \in R^{m_k}} \|f(x_k) - \sum_{j=1}^{m_k} \lambda_j (f(x_{k-m_k+j}) - f(x_{k-m_k+j-1}))\|_2^2;$$

Step 5. *Set*

$$x_{k+1} = g(x_k) + \sum_{j=1}^{m_k-1} \lambda_j [g(x_{k-m_k+j+1}) - g(x_{k-m_k+j})].$$

In this we define the following matrix functions

$$f(Y, Z, M, N) = g(Y, Z, M, N) - (Y, Z, M, N),$$

where $g(Y_k, Z_k, M_k, N_k) = (Y_{k+1}, Z_{k+1}, M_{k+1}, N_{k+1})$ with $Y_{k+1}, Z_{k+1}, M_{k+1}$ and N_{k+1} are computed by (b)–(e) in Algorithm 3.1, and let $f_k = f(Y_k, Z_k, M_k, N_k)$, $g_k = g(Y_k, Z_k, M_k, N_k)$, then Algorithm 3.1 with Anderson acceleration can be described as the following Algorithm 6.2.

Algorithm 6.2. *Algorithm 3.1 with Anderson acceleration to solve problem (3.1).*

Step 1. *Input matrices A, B, C, L and U. Input constant $\varepsilon > 0$, error tolerance $\varepsilon_{out} > 0$, penalty parameter $\alpha > 0$ and integer $m \geq 1$. Choose initial matrices $Y_0, Z_0, M_0, N_0 \in R^{n\times n}$. Let $k =: 0$;*
Step 2. *Compute*

$$X_{k+1} = arg \min_{X\in R^{n\times n}} L_\alpha(X, Y_k, Z_k, M_k, N_k); \tag{6.1}$$

Step 3. *Let $m_k = \min(m, k)$;*
Step 4. *Compute $\lambda_k = (\lambda_1, \lambda_2, \cdots, \lambda_{m_k})^T \in \mathbb{R}^{m_k}$ that solves*

$$\min_{\lambda\in R^{m_k}} \|f_k - \sum_{j=1}^{m_k} \lambda_j (f_{k-m_k+j+1} - f_{k-m_k+j})\|^2;$$

Step 5. *Set*

$$(Y_{k+1}, Z_{k+1}, M_{k+1}, N_{k+1}) = g_k + \sum_{j=1}^{m_k-1} \lambda_j (g_{k-m_k+j+1} - g_{k-m_k-j});$$

Step 6. *If $(\|Y_{k+1} - Y_k\|^2 + \|Z_{k+1} - Z_k\|^2 + \|M_{k+1} - M_k\|^2 + \|N_{k+1} - N_k\|^2)^{1/2} \leq \varepsilon_{out}$, stop. In this case, X_{k+1} is an approximate solution of problem (3.1);*
Step 7. *Let $k =: k + 1$ and go to step 2.*

Table 2. Numerical comparisons between MSADMM and ACADMM.

α=n	MSADMM(γ=1.5)	ACADMM(m=2)	ACADMM(m=10)	ACADMM(m=20)
20	0.0735	0.0991	0.0521	0.0647
40	0.2595	0.3485	0.2186	0.2660
80	1.1866	1.1319	1.0232	1.1069
100	1.7750	2.2081	1.6003	1.7660
150	3.8474	5.2760	3.5700	3.9587
200	7.6133	10.8719	6.9807	7.7318
300	21.2970	28.5379	18.8233	19.9526
400	56.2133	74.3192	44.8087	46.5275
500	98.0542	130.2326	75.6044	80.1480
600	157.7573	208.1124	125.2842	133.4549

Algorithm 6.2 (ACADMM) is m-step iterative method, that is, the current iterates is obtained by the linear combination of the previous m steps. Furthermore, the combination coefficients of the

linear combination are modified at each iteration steps. Compared to ACADMM, Algorithm 3.2 (MSADMM) is two-step iterative method and the combination coefficients of the linear combination are fixed at each iteration steps. The convergence speed of ACADMM is directly related to the penalty parameter α and the backtracking step m. The selection of the penalty parameter α is the same as ADMM since ACADMM's iterates are corrected by ADMM's iterates. For the backtracking step m, as showed in Table 2 (the average computing time of 10 tests) and Figure 10, aggressive values such as $m = 10$ are often preferred (in this case, ACADMM is more efficient than MSADMM). However, how to select the best backtracking step m is an important problem which should be studied in near future.

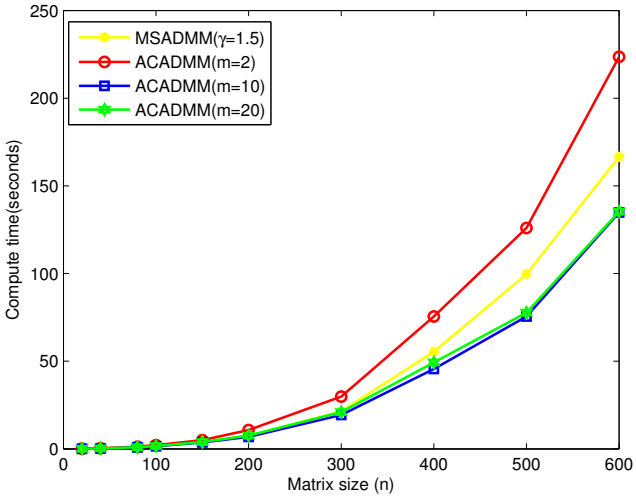

Figure 10. Numerical comparisons between MSADMM and ACADMM.

7. Conclusions

In this paper, the multiple constraint least squares solution of the Sylvester equation $AX + XB = C$ is discussed. The necessary and sufficient conditions for the existence of solutions to the considered problem are given (Theorem 2.1). MSADMM to solve the considered problem is proposed and some convergence results of the proposed algorithm are proved (Theorem 4.1 and Theorem 4.2). Problems which should be studied in the near future are listed. Numerical experiments show that MSADMM with a suitable correction factor γ is more effective than ADMM (See Table 1 and Figure 10), and ACADMM with a suitable backtracdking step m is the most effective of ADMM, MSADMM and ACADMM (See Table 2 and Figure 10).

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by National Natural Science Foundation of China (grant number 11961012) and Special Research Project for Guangxi Young Innovative Talents (grant number AD20297063).

Conflict of interest

The authors declare no competing interests.

References

1. R. Bhatia, P. Rosenthal, How and why to solve the operator equation $AX - XB = Y$, *Bull. London Math. Soc.*, **29** (1997), 1–21. https://doi.org/10.1112/S0024609396001828
2. G. H. Golub, C. F. Van Loan, *Matrix computations*, 3 Eds., Johns Hopkins University Press, Baltimore, Maryland, 1996.
3. V. Sima, *Algorithms for linear quadratic optimization*, Pure and Applied Mathematics, New York: Marcel Dekker, Inc., 1996.
4. B. Datta, *Numerical methods for linear control systems*, Design and Analysis, Academic Press, 2004. https://doi.org/10.1016/B978-0-12-203590-6.X5000-9
5. A. Locatelli, *Optimal control: an introduction*, Birkhäuser Basel, 2001.
6. R. W. Aldhaheri, Model order reduction via real Schur-form decomposition, *Int. J. Control*, **53** (1991), 709–716. https://doi.org/10.1080/00207179108953642
7. A. C. Antoulas, *Approximation of large scale dynamical systems*, Society for Industrial and Applied Mathematics, Philadelphia, 2005.
8. U. Baur, P. Benner, Cross-gramian based model reduction for data-sparse systems, *Electron. Trans. Numer. Anal.*, **31** (2008), 256–270.
9. D. C. Sorensen, A. C. Antoulas, The Sylvester equation and approximate balanced reduction, *Linear Algebra Appl.*, **351-352** (2002), 671–700. https://doi.org/10.1016/S0024-3795(02)00283-5
10. Y. Sun, C. Wu, S. Zhao, Applications of the Sylvester equation for the lattice BKP system, *Theor. Math. Phys.*, **214** (2023), 354–368. https://doi.org/10.1134/S0040577923030042
11. A. Altavilla, C. de Fabritiis, Equivalence of slice semi-regular functions via Sylvester operators, *Linear Algebra Appl.*, **607** (2020), 151–189. https://doi.org/10.1016/j.laa.2020.08.009
12. C. H. Choi, A. J. Laub, Efficient matrix-valued algorithm for solving stiff Riccati differential equations, *IEEE Trans. Autom. Control*, **35** (1989), 770–776. https://doi.org/10.1109/9.57015
13. L. Dieci, Numerical integration of the differential Riccati equation and some related issues, *SIAM J. Numer. Anal.*, **29** (1992), 781–815. https://doi.org/10.1137/0729049
14. W. H. Enright, Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations, *ACM Trans. Math. Software*, **4** (1978), 127–136. https://doi.org/10.1145/355780.355784

15. R. H. Bartels, G. W. Stewart, Algorithm 432: the solution of the matrix equation $AX - BX = C$, *Commun. ACM*, **15** (1972), 820–826. https://doi.org/10.1145/361573.361582

16. G. Golub, S. Nash, C. F. Van Loan, A Hessenberg-Schur method for the matrix equation $AX+XB = C$, *IEEE Trans. Automat. Control*, **24** (1979), 909–913. https://doi.org/10.1109/TAC.1979.1102170

17. J. D. Roberts, Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, *Int. J. Control*, **32** (1980), 677–687. https://doi.org/10.1080/00207178008922881

18. R. A. Smith, Matrix equation $XA + BX = C$, *SIAM J. Appl. Math.*, **16** (1968), 198–201. https://doi.org/10.1137/0116017

19. D. Calvetti, L. Reichel, Application of ADI iterative methods to the restoration of noisy images, *SIAM J. Matrix Anal. Appl.*, **17** (1996), 165–186. https://doi.org/10.1137/S0895479894273687

20. N. Truhar, R. C. Li, On ADI method for Sylvester equations, *Technical Report 2008-02*, Department of Mathematics, University of Texas at Arlington, 2008.

21. E. L. Wachspress, Trail to a Lyapunov equation solver, *Comput. Math. Appl.*, **55** (2008), 1653–1659. https://doi.org/10.1016/j.camwa.2007.04.048

22. E. Wachspress, Adi iteration parameters for solving Lyapunov and Sylvester equations, *Technical Report*, March, 2009.

23. P. Benner, Factorized solution of Sylvester equations with applications in control, *Processing of the 16th International Symposium on Mathematical Theory of Network and Systems (MTNS 2004)*, Leuven, Belgium, 2004.

24. U. Baur, Low rank solution of data sparse Sylvester equations, *Numer. Linear Algebra Appl.*, **15** (2008), 837–851. https://doi.org/10.1002/nla.605

25. M. Konstantinov, D. W. Gu, V. Mehrmann, P. Petkov, *Perturbation theory for matrix equations*, Elsevier, 2003.

26. N. J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, Philadelphia, 1996.

27. G. W. Stewart, J. G. Sun, *Matrix perturbation theory*, Academic Press, Harcourt Brace Jovanovich, 1990.

28. G. B. Dantzig, Deriving a utility function for the economy, *Technical Report SOL 85-6R*, Department of Operations Research, Stanford University, Stanford, CA 1985.

29. H. Hu, I. Olkin, A numerical procedure for finding the positive definite matrix closest to a patterned matrix, *Stat. Probab. Lett.*, **12** (1991), 511–515. https://doi.org/10.1016/0167-7152(91)90006-D

30. M. Monsalve, J. Moreno, R. Escalante, M. Raydan, Selective alternating projections to find the nearest SDD^+ matrix, *Appl. Math. Comput.*, **145** (2003), 205–220. https://doi.org/10.1016/S0096-3003(02)00478-2

31. S. Q. Ma, Alternating proximal gradient method for convex minimization, *J. Sci. Comput.*, **68** (2016), 546–572. https://doi.org/10.1007/s10915-015-0150-0

32. D. P. Bertsekas, J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods,* Prentice-Hall, Inc., 1989.

33. J. Eckstein, D. P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Math. Program.*, **55** (1992), 293–318. https://doi.org/10.1007/BF01581204

34. Z. Y. Peng, A matrix LSQR iterative method to solve matrix equation $AXB = C$, *Int. J. Comput. Math.*, **87** (2010), 1820–1830. https://doi.org/10.1080/00207160802516875

35. D. G. Anderson, Iterative procedures for nonlinear integral equations, *J. ACM*, **12** (1965), 547–560. https://doi.org/10.1145/321296.321305

36. H. Fang, Y. Saad, Two classes of multisecant methods for nonlinear acceleration, *Numer. Linear Algebra Appl.*, **16** (2009), 197–221. https://doi.org/10.1002/nla.617

37. H. F. Walker, P. Ni, Anderson acceleration for fixed-point iterations, *SIAM J. Numer. Anal.*, **49** (2011), 1715–1735. https://doi.org/10.1137/10078356X

38. N. J. Higham, N. Strabic, Anderson acceleration of the alternating projections method for computing the nearest correlation matrix, *Numer. Algor.*, **72** (2016), 1021–1042. https://doi.org/10.1007/s11075-015-0078-3

AIMS Press