

A CLASS OF SMITH-LIKE ITERATION METHOD FOR M-MATRIX SYLVESTER EQUATIONS

GUAN Jin-rui, WANG Zhi-xin, REN Fu-jiao

(School of Mathematics and Statistics, Taiyuan Normal University, Shanxi 030619, China)

Abstract: This paper considers numerical methods for solving M-matrix Sylvester equations, which are widely encountered in many fields of scientific computing and engineering applications. Based on the properties of M-matrix and the idea of Smith method, a class of Smith-like iteration method is proposed to solve M-matrix Sylvester equations, and convergence analysis of the new method is given. Numerical experiments show that the proposed method is feasible and is effective under certain conditions.

Keywords: Sylvester equation; M-matrix; Smith method

2010 MR Subject Classification: 65F30; 65F99; 15B99

Document code: A

Article ID: 0255-7797(2023)05-0389-09

1 Introduction

Sylvester equations play an important role in many fields of scientific computing and engineering applications, such as control theory, linear system theory, numerical solution of differential equations, optimization theory, signal processing, statistics and so on (see [1-4]). In this paper, we consider Sylvester equation of the form

$$AX + XB = C, \quad (1.1)$$

where $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{n \times n}$ are M-matrices with at least one of them be nonsingular, and $C \in \mathbb{R}^{m \times n}$ is a nonnegative matrix. This kind of Sylvester equation is called M-matrix Sylvester equation (see [5]). Under the above conditions, equation (1.1) has a unique nonnegative matrix solution. M-matrix Sylvester equation appears frequently in iterative methods for the study of nonsymmetric algebraic Riccati equations, while M-matrix Lyapunov equation, i.e., $B = A^T$, arises in positive systems (see [6]).

The following are some definitions and lemmas, which are mainly from [7].

Let $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, then A is called a nonnegative matrix if $a_{ij} \geq 0$ for all i, j . For $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{m \times n}$, we write $A \geq B$ ($A > B$), if $a_{ij} \geq b_{ij}$ ($a_{ij} > b_{ij}$) for all i, j . A square matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is called a Z-matrix, if $a_{ij} \leq 0$ for all $i \neq j$. Any Z-matrix A

* Received date: 2022-06-20

Accepted date: 2022-09-23

Foundation item: Supported by National Natural Science Foundation of China (12001395), Natural Science Foundation of Shanxi province (201901D211423).

Biography: Guan Jinrui(1985-), male, Shanxi, associate professor, major in numerical linear algebra and its applications. E-mail: guanjinrui2012@163.com.

can be written as $A = sI - B$, where s is a scalar and B is a nonnegative matrix. In addition, A is called an M-matrix if $s \geq \rho(B)$, where $\rho(B)$ is the spectral radius of B . In particular, A is called a nonsingular M-matrix if $s > \rho(B)$ and singular M-matrix if $s = \rho(B)$.

Lemma 1.1 Let $A \in \mathbb{R}^{n \times n}$ be a Z-matrix. Then the following three statements are equivalent:

- (1) A is a nonsingular M-matrix;
- (2) $A^{-1} \geq 0$;
- (3) $Av > 0$ for some vectors $v > 0$.

Lemma 1.2 Let A and B be Z-matrices. If A is a nonsingular M-matrix and $A \leq B$, then B is also a nonsingular M-matrix. In particular, for any $\alpha \geq 0$, $B = \alpha I + A$ is a nonsingular M-matrix.

The rest of the paper is organized as follows. In Section 2, we review some iteration methods for solving M-matrix Sylvester equations. In Section 3, we propose a class of Smith-like iteration method and give convergence analysis of it. In Section 4, we use some numerical examples to show the feasibility and effectiveness of the new method. Conclusion is given in Section 5.

2 Some Existing Methods

Due to the wide applications of Sylvester equations, many efficient numerical methods have been proposed for solving them. The basic method is Bartels-Stewart algorithm, which is very effective for small-scale problems (see [8]). However, when the problem scale is large, iterative methods are more effective than direct methods. The common iterative methods for solving Sylvester equation include Smith method [9], alternating direction method [10], Krylov subspace method [11], and etc. For some recent development, see [12-17] for details.

Smith method is a classical iterative method for solving Sylvester equations, which is quadratically convergent. For M-matrix Sylvester equation, it is constructed as follows (see [9,18] for details). For a fixed scalar $\mu > 0$, equation (1.1) can be written as

$$(\mu I + A)X(\mu I + B) - (\mu I - A)X(\mu I - B) = 2\mu C,$$

thus we have $X = X_0 + E_0 X F_0$, where

$$X_0 = 2\mu(\mu I + A)^{-1}C(\mu I + B)^{-1},$$

$$E_0 = (\mu I + A)^{-1}(\mu I - A), \quad F_0 = (\mu I - B)(\mu I + B)^{-1}.$$

The solution of (1.1) admits the following series expansion $X = \sum_{i=0}^{\infty} E_0^i X_0 F_0^i$ which can be quickly approximated by $X_{k+1} = X_k + E_0^{2^k} X_k F_0^{2^k}$, $k \geq 0$.

Algorithm 1: Smith method

- (1) Choose $\mu = \max\{a_{ii}, b_{ii}\}$;
- (2) Compute

$$X_0 = 2\mu(\mu I + A)^{-1}C(\mu I + B)^{-1},$$

$$E_0 = (\mu I + A)^{-1}(\mu I - A), \quad F_0 = (\mu I - B)(\mu I + B)^{-1};$$

(3) For $k = 0, 1, \dots$ until convergence, compute

$$X_{k+1} = X_k + E_k X_k F_k, \quad E_{k+1} = E_k^2, \quad F_{k+1} = F_k^2.$$

Recently, Wang et al proposed an alternating directional Smith method in [18], which is differ to Smith method only in the initial settings. It is constructed as follows (see [18] for details). Choose $\alpha, \beta > 0$, and write equation (1.1) as $(\beta I + A)X(\alpha I + B) - (\alpha I - A)X(\beta I - B) = (\alpha + \beta)C$, thus we have $X = X_0 + E_0 X F_0$, where

$$X_0 = (\alpha + \beta)(\beta I + A)^{-1}C(\alpha I + B)^{-1},$$

$$E_0 = (\beta I + A)^{-1}(\alpha I - A), \quad F_0 = (\beta I - B)(\alpha I + B)^{-1}.$$

The solution of (1.1) admits the following series expansion

$$X = \sum_{i=0}^{\infty} E_0^i X_0 F_0^i$$

which can be quickly approximated by

$$X_{k+1} = X_k + E_0^{2^k} X_k F_0^{2^k}, \quad k \geq 0.$$

The rest of the method is similar to Smith method.

Algorithm 2: Alternating directional Smith method

(1) Choose $\alpha = \max\{a_{ii}\}$, $\beta = \max\{b_{ii}\}$;

(2) Compute

$$X_0 = (\alpha + \beta)(\beta I + A)^{-1}C(\alpha I + B)^{-1},$$

$$E_0 = (\beta I + A)^{-1}(\alpha I - A), \quad F_0 = (\beta I - B)(\alpha I + B)^{-1};$$

(3) For $k = 0, 1, \dots$ until convergence, compute

$$X_{k+1} = X_k + E_k X_k F_k, \quad E_{k+1} = E_k^2, \quad F_{k+1} = F_k^2.$$

Compared with Smith method, the alternating directional Smith method has smaller convergence factor, so it usually converges faster. However, the alternating directional Smith method still has room for improvement. In this paper, we will propose a class of Smith-like method to solve M-matrix Sylvester equations. Compared with Smith method and alternating directional Smith method, the new method has less initial computation counts and thus has better numerical behaviours under certain conditions.

3 A Class of Smith-like Iteration Method

In this section, based on the properties of M-matrix and the idea of Smith method, we propose a class of Smith-like iteration method to solve M-matrix Sylvester equations, and give convergence analysis of the new method.

Choose parameter $\alpha > 0$, and write equation (1.1) as $X(\alpha I + B) = (\alpha I - A)X + C$, then we have $X = X_0 + E_0 X F_0$, where

$$X_0 = C(\alpha I + B)^{-1}, \quad E_0 = \alpha I - A, \quad F_0 = (\alpha I + B)^{-1}. \quad (3.1)$$

The solution can be written as

$$X = \sum_{i=0}^{\infty} E_0^i X_0 F_0^i.$$

As in the Smith method, we can approximate the solution by the following iteration:

$$\begin{cases} X_{k+1} = X_k + E_k X_k F_k, \\ E_{k+1} = E_k^2, \quad F_{k+1} = F_k^2, \end{cases}, \quad k = 0, 1, 2, \dots \quad (3.2)$$

Compared with Smith method and alternating directional Smith method, the above method only needs to calculate the inverse of one matrix in the initial condition (3.1), so the amount of computation is small. In the bellow, we give the convergence analysis of the method (3.1)-(3.2).

Lemma 3.1 For equation (1.1), if the parameter α satisfies $\alpha \geq \{a_{ii}\}$, then X_0, E_0, F_0 defined in (3.1) are all nonnegative.

Proof Since A is an M-matrix, and $\alpha \geq \{a_{ii}\}$, we know that $E_0 = \alpha I - A \geq 0$. On the other hand, since B is an M-matrix, and $\alpha > 0$, we can conclude from Lemma 1.2 that $\alpha I + B$ is a nonsingular M-matrix. Thus from Lemma 1.1, we have $F_0 = (\alpha I + B)^{-1} \geq 0$. Similarly, we have $X_0 = C(\alpha I + B)^{-1} \geq 0$.

Theorem 3.1 For equation (1.1), if the parameter α satisfies $\alpha \geq \{a_{ii}\}$, then the sequence $\{X_k\}$ generated by (3.2) is nonnegative, monotonically increasing and converges to X . In addition, the convergent rate is quadratic, and the convergent factor is given by

$$\limsup_{k \rightarrow \infty} \sqrt[2^k]{\|X - X_k\|} \leq \frac{\alpha - \lambda_{\min}(A)}{\alpha + \lambda_{\min}(B)},$$

where $\lambda_{\min}(A)$, $\lambda_{\min}(B)$ are the minimum nonnegative eigenvalues of A, B respectively.

Proof From Lemma 3.1 and the iteration (3.2), it is easy to verify that the sequence $\{X_k\}$ is nonnegative, monotonically increasing. In addition, we can conclude that

$$X_k = \sum_{i=0}^{2^k-1} E_0^i X_0 F_0^i.$$

Thus we have

$$X - X_k = \sum_{i=2^k}^{\infty} E_0^i X_0 F_0^i = E_0^{2^k} \left(\sum_{i=0}^{\infty} E_0^i X_0 F_0^i \right) F_0^{2^k} = E_0^{2^k} X F_0^{2^k}.$$

Taking norm on both sides and noting that $\lim_{k \rightarrow \infty} \sqrt[2^k]{\|A^k\|} = \rho(A)$, we have

$$\limsup_{k \rightarrow \infty} \sqrt[2^k]{\|X - X_k\|} \leq \rho(E_0) \cdot \rho(F_0).$$

Since E_0, F_0 are nonnegative, it is easy to conclude that

$$\rho(E_0) = \alpha - \lambda_{\min}(A), \quad \rho(F_0) = \frac{1}{\alpha + \lambda_{\min}(B)}.$$

In addition, since A and B are M-matrices with at least one of them be nonsingular, we have $\frac{\alpha - \lambda_{\min}(A)}{\alpha + \lambda_{\min}(B)} < 1$. Thus the convergent rate of (3.2) is quadratic.

Corollary 3.1 Let the assumptions be as in Theorem 3.1, then the optimal parameter in (3.1)-(3.2) is $\alpha = \{a_{ii}\}$.

Proof Since the convergent factor is given by $\frac{\alpha - \lambda_{\min}(A)}{\alpha + \lambda_{\min}(B)}$, it is clear that the smaller the parameter be, and the smaller the convergence factor will be. Thus the optimal parameter is $\alpha = \{a_{ii}\}$.

Similarly, if we choose parameter $\beta > 0$, equation (1.1) can be written as

$$(\beta I + A)X = X(\beta I - B) + C,$$

and we can have

$$X = X_0 + E_0 X F_0,$$

where

$$X_0 = (\beta I + A)^{-1}C, \quad E_0 = (\beta I + A)^{-1}, \quad F_0 = \beta I - B. \quad (3.3)$$

The solution can be written as

$$X = \sum_{i=0}^{\infty} E_0^i X_0 F_0^i.$$

And similarly, we can reach to the following iteration:

$$\begin{cases} X_{k+1} = X_k + E_k X_k F_k, \\ E_{k+1} = E_k^2, \quad F_{k+1} = F_k^2, \end{cases}, \quad k = 0, 1, 2, \dots \quad (3.4)$$

We have similar convergence results for the above iterative method (3.3)-(3.4).

Theorem 3.2 For equation (1.1), if the parameter β satisfies $\beta \geq \{b_{ii}\}$, then the sequence $\{X_k\}$ generated by (3.4) is nonnegative, monotonically increasing and converges to X . In addition, the convergent rate is quadratic, and the convergent factor is

$$\limsup_{k \rightarrow \infty} \sqrt[2^k]{\|X - X_k\|} \leq \frac{\beta - \lambda_{\min}(B)}{\beta + \lambda_{\min}(A)},$$

where $\lambda_{\min}(A)$, $\lambda_{\min}(B)$ are the minimum nonnegative eigenvalues of A, B respectively.

Corollary 3.2 Let the assumptions be as in Theorem 3.2, then the optimal parameter in (3.3)-(3.4) is $\beta = \{b_{ii}\}$.

From the convergence analysis of the above iterative methods (3.2) and (3.4), it can be seen that the convergence factors of the two iterative methods are generally different. In order to achieve faster convergence in practical calculation, we consider both the numerical effects of the two iterative methods and propose the following algorithm.

Algorithm 3: Smith-like iteration method

- (1) Compute $\alpha = \max\{a_{ii}\}$, $\beta = \max\{b_{ii}\}$, where a_{ii}, b_{ii} are the diagonal entries of A, B respectively;
- (2) If $\alpha \leq \beta$, compute according to (3.1)-(3.2);
- (3) If $\alpha > \beta$, compute according to (3.3)-(3.4).

4 Numerical Experiments

In this section we use several examples to show the feasibility and effectiveness of the Smith-like method. We will compare the numerical behaviours of the Smith method, the alternating directional Smith method, and the Smith-like iteration method, denoted by Alg 1, Alg 2, Alg 3 respectively. The numerical results are presented in terms of the numbers of iterations (IT), CPU time (CPU) in seconds and the residue (RES), where

$$RES := \frac{\|AX + XB - C\|_{\infty}}{\|C\|_{\infty}}.$$

In our implementations all iterations are performed in MATLAB (version R2012a) on a personal computer and are terminated when the current iterate satisfies $RES < 10^{-12}$.

Example 4.1 Consider equation (1.1), where

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Numerical results are reported in Table 1 for this example. From Table 1, we can conclude that all the three methods can compute the solution as desired accuracy, and all have better numerical behaviours. In addition, the number of iterations required by Algorithm 2 is slightly smaller, and the CPU time of Algorithm 3 is slightly less.

Table 1 Numerical Results of Example 4.1

Method	IT	CPU	RES
Alg 1	6	0.0005	4.4409e-16
Alg 2	5	0.0005	1.1102e-15
Alg 3	6	0.0004	8.8818e-16

Example 4.2 Consider equation (1.1), where

$$A = \begin{pmatrix} 102 & -100 \\ -100 & 102 \end{pmatrix}, \quad B = \begin{pmatrix} 3 & -1 \\ -1 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Numerical results are reported in Table 2. For this problem, all the three methods can compute the solution as desired accuracy, and the numerical behaviours of the latter two methods are better. In particular, the CPU time of Algorithm 3 is slightly less than that of Algorithm 2.

Table 2 Numerical Results of Example 4.2

Method	IT	CPU	RES
Alg 1	9	0.0006	3.8303e-15
Alg 2	5	0.0005	4.7740e-15
Alg 3	5	0.0004	4.6629e-15

Example 4.3 Consider equation (1.1) with

$$A = \begin{pmatrix} 2 & -1 & & & \\ & 2 & -1 & & \\ & & \ddots & \ddots & \\ & & & 2 & -1 \\ -1 & & & & 2 \end{pmatrix} \in \mathbb{R}^{n \times n}, \quad B = \omega A, \quad C = I.$$

Here we take $n = 100$. For different values of parameter ω , the numerical results are shown in Table 3. It can be seen from the table that the numerical effects of the three methods are all well. In particular, when $\alpha = \max\{a_{ii}\}$ and $\beta = \max\{b_{ii}\}$ differ greatly, Algorithm 3 requires the least CPU time.

Table 3 Numerical Results of Example 4.3

ω	Method	IT	CPU	RES
1	Alg 1	4	3.7869	5.3966e-16
	Alg 2	4	3.7408	5.3966e-16
	Alg 3	5	3.4153	5.3966e-16
10	Alg 1	5	5.0133	3.2582e-16
	Alg 2	4	4.6787	1.0607e-16
	Alg 3	4	4.2557	1.0762e-16
100	Alg 1	5	5.8128	4.2082e-16
	Alg 2	3	3.8218	2.3106e-17
	Alg 3	3	3.4193	1.2174e-16

Example 4.4 Consider equation (1.1) with

$$A = \begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} n+1 & -1 & \cdots & -1 & -1 \\ -1 & n+1 & \cdots & -1 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \cdots & n+1 & -1 \\ -1 & -1 & \cdots & -1 & n+1 \end{pmatrix},$$

and $C = I$. For different values of n , the numerical results are shown in Table 4. It can be seen that the numerical effects of the three methods are all well. In particular, Algorithm 3 requires the least CPU time.

Table 4 Numerical Results of Example 4.4

n	Method	IT	CPU	RES
50	Alg 1	8	0.0227	8.9882e-14
	Alg 2	5	0.0195	3.3946e-14
	Alg 3	5	0.0145	1.8756e-13
100	Alg 1	9	0.0395	1.1469e-13
	Alg 2	5	0.0224	1.1316e-13
	Alg 3	5	0.0219	2.2306e-13
200	Alg 1	10	0.1071	1.6718e-13
	Alg 2	5	0.0749	2.4595e-13
	Alg 3	5	0.0609	2.5491e-13
400	Alg 1	11	0.7105	5.3530e-13
	Alg 2	5	0.3531	6.2832e-13
	Alg 3	5	0.2374	6.7059e-13
600	Alg 1	12	1.9798	9.4068e-13
	Alg 2	6	1.0114	9.1798e-13
	Alg 3	6	0.7328	8.2318e-13

From the above four examples, it can be seen that Algorithm 3 is feasible. In particular, when $\alpha = \max\{a_{ii}\}$ and $\beta = \max\{b_{ii}\}$ differ greatly, Algorithm 3 requires the lest CPU time, so it is also effective.

5 Conclusions

In this paper, the numerical solution of M-matrix Sylvester equations is studied, and a class of Smith-like iteration method is proposed to solve the equation. Convergence analysis and numerical examples show that the new method is feasible. In addition, it is effective under certain conditions.

References

- [1] Gajic Z, Qureshi M T J. Lyapunov matrix equation in system stability and control [M]. New York: Academic Press, 1995.
- [2] Datta B. Numerical methods for linear control systems [M]. New York: Academic Press, 2004.
- [3] Bini D A, Iannazzo B, Meini B. Numerical solution of algebraic Riccati equations [M]. Philadelphia: SIAM, 2012.
- [4] Simoncini V. Computational methods for linear matrix equations [J]. SIAM Rev., 2016, 58(3): 377–441.
- [5] Xue Jungong, Xu Shufang, Li Rencang. Accurate solutions of M-matrix Sylvester equations [J]. Numer. Math., 2012, 120(4): 639–670.
- [6] Virnik E. Analysis of positive descriptor systems [D]. Berlin: Technischen Universitat Berlin, 2008.

- [7] Berman A, Plemmons R J. Nonnegative matrices in the mathematical sciences [M]. New York: Academic Press, 1994.
- [8] Bartels R H, Stewart G W. The solution of the matrix equation $AX-BX=C$ [J]. Commun. ACM, 1972, 8(2): 820–826.
- [9] Smith R A. Matrix equation $XA+BX=C$ [J]. SIAM J. Appl. Math., 1968, 16(1): 198–201.
- [10] Benner P, Li Rencang, Truhar N. On the ADI method for Sylvester equations [J]. J. Comput. Appl. Math., 2009, 233(4): 1035–1045.
- [11] Bai Zhongzhi. On Hermitian and skew-Hermitian splitting iteration methods for continuous Sylvester equations [J]. J. Comput. Math., 2011, 29(2): 185–198.
- [12] Wang Xiang, Dai Lin, Liao Dan. A modified gradient based algorithm for solving Sylvester equations [J]. Appl. Math. Comput, 2012, 218(9): 5620–5628.
- [13] Wang Qingwen, He Zhuoheng. Systems of coupled generalized Sylvester matrix equations [J]. Automatica, 2014, 50(11): 2840–2844.
- [14] Huang Baohua, Ma Changfeng. Extending GCR algorithm for the least squares solutions on a class of Sylvester matrix equations [J]. Numerical Mathematics: Theory Methods and Applications, 2018, 11(1): 140–159.
- [15] Huang Baohua, Ma Changfeng. Finite iterative algorithm for the symmetric periodic least squares solutions of a class of periodic Sylvester matrix equations [J]. Numer. Alg., 2019, 81(1): 377–406.
- [16] Zhang Juan, Kang Huihui. The generalized modified Hermitian and skew-Hermitian splitting method for the generalized Lyapunov equation [J]. International Journal of Control, Automation and Systems, 2021, 19(1): 339–349.
- [17] Zhang Juan, Luo Xiao. Gradient-based optimization algorithm for solving Sylvester matrix equation [J]. Mathematics, 2022, 10(7): 1040.
- [18] Wang Weiguo, Wang Weichao, Li Rencang. Alternating-directional doubling algorithm for M-matrix algebraic Riccati equations [J]. SIAM J. Matrix Anal. Appl., 2012, 33(1): 170–194.

M- 矩阵 Sylvester 方程的一类 Smith-like 迭代法

关晋瑞, 王志欣, 任孚蛟

(太原师范学院数学与统计学院, 山西 晋中 030619)

摘要: 本文研究了 M- 矩阵 Sylvester 方程的数值解法, 这类矩阵方程广泛出现在科学计算和工程应用的许多领域. 利用 M- 矩阵的性质和 Smith 方法的思想, 提出了一类 Smith-like 迭代法以求解 M- 矩阵 Sylvester 方程, 并给出了新方法的收敛性分析. 数值实验表明, 新方法是可行的, 而且在一定条件下也是较为有效的.

关键词: Sylvester 方程; M- 矩阵; Smith 方法

MR(2010) 主题分类号: 65F30; 65F99; 15B99

中图分类号: O241.6