

## Monotone convergence of Newton-like methods for $M$ -matrix algebraic Riccati equations

Chun-Hua Guo

Received: date / Accepted: date

**Abstract** For the algebraic Riccati equation whose four coefficient matrices form a nonsingular  $M$ -matrix or an irreducible singular  $M$ -matrix  $K$ , the minimal nonnegative solution can be found by Newton's method and the doubling algorithm. When the two diagonal blocks of the matrix  $K$  have both large and small diagonal entries, the doubling algorithm often requires many more iterations than Newton's method. In those cases, Newton's method may be more efficient than the doubling algorithm. This has motivated us to study Newton-like methods that have higher-order convergence and are not much more expensive each iteration. We find that the Chebyshev method of order three and a two-step modified Chebyshev method of order four can be more efficient than Newton's method. For the Riccati equation, these two Newton-like methods are actually special cases of the Newton–Shamanskii method. We show that, starting with zero initial guess or some other suitable initial guess, the sequence generated by the Newton–Shamanskii method converges monotonically to the minimal nonnegative solution. We also explain that the Newton-like methods can be used to great advantage when solving some Riccati equations involving a parameter.

**Keywords** Algebraic Riccati equation ·  $M$ -matrix · Minimal nonnegative solution · Newton-like method · Chebyshev's method · Monotone convergence

**Mathematics Subject Classification (2000)** 15A24 · 65F30 · 65H10

---

This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada

Chun-Hua Guo  
Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2,  
Canada  
E-mail: chguo@math.uregina.ca

## 1 Introduction

We consider the algebraic Riccati equation (ARE)

$$\mathcal{R}(X) = XCX - XD - AX + B = 0, \quad (1)$$

where  $A, B, C, D$  are real matrices of sizes  $m \times m, m \times n, n \times m, n \times n$ , respectively, and the matrix

$$K = \begin{bmatrix} D & -C \\ -B & A \end{bmatrix} \quad (2)$$

is a nonsingular  $M$ -matrix or an irreducible singular  $M$ -matrix. Such an ARE has been called an  $M$ -matrix ARE in [30] and the acronym MARE is used there. We will adopt this acronym in this paper. Some relevant definitions are as follows. For any matrices  $A, B \in \mathbb{R}^{m \times n}$ , we write  $A \geq B$  ( $A > B$ ) if  $a_{ij} \geq b_{ij}$  ( $a_{ij} > b_{ij}$ ) for all  $i, j$ . A real square matrix  $A$  is called a  $Z$ -matrix if all its off-diagonal elements are nonpositive. Note that any  $Z$ -matrix  $A$  can be written as  $sI - B$  with  $B \geq 0$ . A  $Z$ -matrix  $A$  is called an  $M$ -matrix if  $s \geq \rho(B)$ , where  $\rho(\cdot)$  is the spectral radius; it is a singular  $M$ -matrix if  $s = \rho(B)$  and a nonsingular  $M$ -matrix if  $s > \rho(B)$ .

The MARE (1) has applications in transport theory and Markov models [21, 25]. The solution of practical interest is the minimal nonnegative solution. The equation has been studied in many papers [1, 4, 6, 8, 9, 11–18, 20–22, 29, 30] and has been a main topic in the recent monograph [5].

For application to Markov models, the case of primary interest is the one where  $K$  is an irreducible singular  $M$ -matrix with zero row sums. When  $K$  is an irreducible singular  $M$ -matrix, there are positive vectors  $u_1, v_1 \in \mathbb{R}^n$  and  $u_2, v_2 \in \mathbb{R}^m$  such that

$$M(v_1^T \ v_2^T)^T = 0, \quad (u_1^T \ u_2^T)M = 0, \quad (3)$$

and the vectors  $(v_1^T \ v_2^T)$  and  $(u_1^T \ u_2^T)$  are each unique up to a scalar multiple.

Some properties of the MARE (1) are summarized below. See [11, 12, 14] for more details.

**Theorem 1** *The MARE (1) has a minimal nonnegative solution  $S$ . If  $K$  is irreducible, then  $S > 0$  and  $A - SC$  and  $D - CS$  are irreducible  $M$ -matrices. If  $K$  is a nonsingular  $M$ -matrix, then  $A - SC$  and  $D - CS$  are nonsingular  $M$ -matrices. If  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 \neq u_2^T v_2$ , then one of the matrices  $A - SC$  and  $D - CS$  is a singular  $M$ -matrix and the other is a nonsingular  $M$ -matrix. If  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ , then the matrices  $A - SC$  and  $D - CS$  are both singular  $M$ -matrices.*

The case where  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$  will be called a critical case. In this case, the matrix

$$I \otimes (A - SC) + (D - CS)^T \otimes I \quad (4)$$

is a singular  $M$ -matrix, where  $\otimes$  is the Kronecker product. For the non-critical case (where  $K$  is a nonsingular  $M$ -matrix or an irreducible singular  $M$ -matrix with  $u_1^T v_1 \neq u_2^T v_2$ ), the matrix (4) is a nonsingular  $M$ -matrix.

The dual equation of (1):

$$YBY - YA - DY + C = 0. \quad (5)$$

is also an  $M$ -matrix ARE [11], and its minimal nonnegative solution will be denoted by  $\widehat{S}$ .

A number of iterative methods for finding the minimal solution of the MARE (1) are already available. They include a class of basic fixed-point iterations [11,16], Newton's method [11,14,16], inexact Newton methods [9], and different versions of the doubling algorithm [6,18,29]. For the non-critical case, the convergence of the basic fixed-point iterations is linear while the convergence of Newton's method and the doubling algorithm is quadratic. In most cases, the doubling algorithm is more efficient than Newton's method. However, when the two diagonal blocks of the matrix  $K$  have both large and small diagonal entries, the doubling algorithm often requires many more iterations than Newton's method. In those cases, Newton's method may be more efficient than the doubling algorithm even though the computational work for Newton's method is roughly three times that for the doubling algorithm. This has motivated us to study Newton-like methods that have higher-order convergence and are not much more expensive each iteration.

## 2 Newton-like methods

The Riccati function  $\mathcal{R}$  in (1) is a mapping from  $\mathbb{R}^{m \times n}$  into itself. The Fréchet derivative of  $\mathcal{R}$  at a matrix  $X$  is a linear map  $\mathcal{R}'_X : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$  given by

$$\mathcal{R}'_X(Z) = -((A - XC)Z + Z(D - CX)). \quad (6)$$

The second derivative at  $X$ ,  $\mathcal{R}''_X : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ , is given by

$$\mathcal{R}''_X(Z_1, Z_2) = Z_1 CZ_2 + Z_2 CZ_1. \quad (7)$$

Newton's method for a solution of (1) is

$$X_{i+1} = X_i - (\mathcal{R}'_{X_i})^{-1} \mathcal{R}(X_i), \quad i = 0, 1, \dots, \quad (8)$$

where the maps  $\mathcal{R}'_{X_i}$  all need to be invertible. By (6), the iteration (8) is equivalent to

$$(A - X_i C)X_{i+1} + X_{i+1}(D - CX_i) = B - X_i CX_i, \quad i = 0, 1, \dots \quad (9)$$

We can also compute  $X_{i+1}$  directly from (8). So  $X_{i+1} = X_i + H_i$ , where  $H_i$  is solved from

$$(A - X_i C)H_i + H_i(D - CX_i) = \mathcal{R}(X_i), \quad i = 0, 1, \dots \quad (10)$$

To assess the computational cost of Newton's method, we make the simplifying assumption that  $m = n$ . The main computational work is with the solution of a Lyapunov equation of the form  $PX + XQ = R$ . It is about  $60n^3$  flops by the Bartels–Stewart algorithm [2], with  $50n^3$  flops used to get the real Schur forms for  $P$  and  $Q$ . (Here we use the operation counts as given in Appendix C of [19].) If we use (9), forming the matrices  $P, Q, R$  requires  $6n^3$  flops; if we use (10), forming the matrices  $P, Q, R$  requires  $8n^3$  flops, noting that  $\mathcal{R}(X) = X(CX - D) - AX + B$ . So the total computational work for the two versions of Newton's method is  $66n^3$  and  $68n^3$ , respectively. The second version tends to provide better accuracy since it implements Newton's method in the form of a correction method. Besides, the residual  $\mathcal{R}(X_i)$  is often used in a stopping criterion and needs to be computed anyway. So we will adopt the second version of Newton's method. Since the computation of Schur forms is the most expensive part of the computation, we would like to reuse these forms for a few more iterations. We thus consider the following Newton-like method, which is also called the Newton–Shamanskii method.

**Algorithm 1** For  $i = 0, 1, \dots$ ,

$$Y_{i,0} = X_i - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(X_i), \quad (11)$$

$$Y_{i,s} = Y_{i,s-1} - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(Y_{i,s-1}), \quad s = 1, \dots, r, \quad (12)$$

$$X_{i+1} = Y_{i,r}. \quad (13)$$

When  $\mathcal{R}'_S$  is invertible, the sequence  $\{X_i\}$  generated by the Newton–Shamanskii method converges to  $S$  if  $X_0$  is sufficiently close to  $S$ , and the convergence order is  $r + 2$  (see [23], for example). Note that the method reduces to Newton's method when  $r = 0$ . However, we can get better efficiency by taking a proper  $r \geq 1$ .

For Newton-like methods for the scalar equation  $f(x) = 0$ , the concept of efficiency index has been introduced in [28]. The efficiency index for an order  $p$  method that requires  $q$  pieces of information each iteration is defined to be  $p^{1/q}$ . Newton's method has second order convergence and requires two pieces of information (one function evaluation and one derivative evaluation), and thus has efficiency index  $\sqrt{2} = 1.4142$ . There have been many papers on designing Newton-like methods with higher efficient indices; see for example [26, 27] and the references therein. However, this measure of efficiency is already rough for the scalar case, and cannot be applied to the Riccati equation for  $n \gg 1$ . We now re-interpret efficiency index using flop counts. For the specific implementation described above, Newton's method requires  $68n^3$  flops. So the efficiency index for Newton's method is still 1.4142 if we regard  $34n^3$  flops as one piece of information. For Newton–Shamanskii method, each step in (12) requires  $16n^3$  flops, since the Schur forms of  $A - X_i C$  and  $D - C X_i$  are already available from the computation in (11). Thus a reasonable efficiency index for Newton–Shamanskii method should be  $e(r) = (r + 2)^{34/(68+16r)}$ . We find that  $e(1) = 1.5600, e(2) = 1.6021, e(3) = 1.6028, e(4) = 1.5865$ , and that  $e(r)$  is monotonically decreasing for  $r \geq 3$ . So we would not use Newton–Shamanskii method for  $r \geq 4$ . Note that  $e(2)$  and  $e(3)$  are very close.

We also note that the computational work for Newton–Shamanskii method for  $r \geq 1$  can be reduced slightly by combining  $Y_{i,0} = X_i - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(X_i)$  and  $Y_{i,1} = Y_{i,0} - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(Y_{i,0})$  into

$$Y_{i,1} = X_i + H_i - \frac{1}{2}(\mathcal{R}'_{X_i})^{-1}\mathcal{R}''_{X_i}(H_i, H_i), \quad (14)$$

where  $H_i = -(\mathcal{R}'_{X_i})^{-1}\mathcal{R}(X_i)$ . This is because by Taylor formula we have

$$\mathcal{R}(Y_{i,0}) = \mathcal{R}(X_i + H_i) = \mathcal{R}(X_i) + \mathcal{R}'_{X_i}(H_i) + \frac{1}{2}\mathcal{R}''_{X_i}(H_i, H_i) = \frac{1}{2}\mathcal{R}''_{X_i}(H_i, H_i).$$

It is easy to see that computing  $Y_{i,1}$  from (14) requires  $82n^3$  flops, instead of the  $84n^3$  flops in the original Newton–Shamanskii method. After this modification, the efficiency index for Newton–Shamanskii method is then  $e(r) = (r+2)^{34/(66+16r)}$  for  $r \geq 1$ . We find that  $e(1) = 1.5770$ ,  $e(2) = 1.6176$ ,  $e(3) = 1.6161$ , and that  $e(r)$  is monotonically decreasing for  $r \geq 2$ . So there is actually no need to use Newton–Shamanskii method for  $r \geq 3$ , in our scenario.

If the Hessenberg–Schur method [10] is used to solve the Lyapunov equation, the flop count is reduced to  $\frac{130}{3}n^3$ , with  $\frac{10}{3}n^3$  flops for one Hessenberg reduction,  $25n^3$  flops for one Schur decomposition, and  $15n^3$  flops for the rest of computations. So one Newton iteration now requires  $\frac{154}{3}n^3$  flops and we would regard  $\frac{77}{3}n^3$  flops as one piece of information. Thus the efficiency index for Newton's method is still 1.4142 and the efficiency index for Newton–Shamanskii method would be  $e(r) = (r+2)^{(77/3)/((154/3-2)+(15+6)r)} = (r+2)^{\frac{77}{148+63r}}$  for  $r \geq 1$ . We find that  $e(1) = 1.4932$ ,  $e(2) = 1.4764$ , and that  $e(r)$  is monotonically decreasing for  $r \geq 1$ . In this scenario, Newton–Shamanskii method with  $r = 1$  has the highest efficiency index.

Unless otherwise stated, we assume that the Bartels–Stewart algorithm is used to solve the Lyapunov equation. Thus, to improve the efficiency of Newton's method, we can use the above variant of Newton–Shamanskii method with  $r = 1$ , which is now

$$X_{i+1} = X_i + H_i - \frac{1}{2}(\mathcal{R}'_{X_i})^{-1}\mathcal{R}''_{X_i}(H_i, H_i) \quad i = 0, 1, \dots \quad (15)$$

This method is also known as Chebyshev's method [7]. To further improve the efficiency of Chebyshev's method, we can use the following modified Chebyshev method.

$$Y_i = X_i + H_i - \frac{1}{2}(\mathcal{R}'_{X_i})^{-1}\mathcal{R}''_{X_i}(H_i, H_i), \quad (16)$$

$$X_{i+1} = Y_i - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(Y_i), \quad i = 0, 1, \dots \quad (17)$$

This modified Chebyshev method is mathematically equivalent to Newton–Shamanskii method with  $r = 2$ .

With the special structure of the MARE (1), we are of course not satisfied with the local convergence of the Newton–Shamanskii method. We would like to prove monotone convergence for suitable initial guesses  $X_0$  that are not necessarily close to the minimal nonnegative solution  $S$ .

### 3 Monotone convergence

For the MARE (1), a monotone convergence result has been proved in [14] for Newton's method. For a special MARE that can be reduced to a vector equation, a monotone convergence result has been proved in [24] when the Newton–Shamanskii method is applied to the vector equation. In this section we will prove a monotone convergence result for Newton–Shamanskii method, for the MARE (1) without any further assumptions. Actually, we will prove monotone convergence even when the integer  $r$  is dependent on  $i$  in (12) of Algorithm 1.

We will need the following result (see [3], for example).

**Theorem 2** *For a  $Z$ -matrix  $A$ , the following are equivalent:*

- (a)  $A$  is a nonsingular  $M$ -matrix.
- (b)  $A^{-1} \geq 0$ .
- (c)  $Av > 0$  for some vector  $v > 0$ .
- (d) All eigenvalues of  $A$  have positive real parts.

The next result is also well-known and follows readily from Theorem 2.

**Lemma 1** *Let  $A$  be a nonsingular  $M$ -matrix. If  $B \geq A$  is a  $Z$ -matrix, then  $B$  is also a nonsingular  $M$ -matrix. Moreover  $B^{-1} \leq A^{-1}$ .*

Note that the first conclusion in the lemma follows from the equivalence of (a) and (c) in Theorem 2, and that the second conclusion follows on writing  $A = s(I - N_A)$  and  $B = s(I - N_B)$  with  $s > 0$  and  $0 \leq N_B \leq N_A$ .

For the MARE (1),  $K$  in (2) is a nonsingular  $M$ -matrix or an irreducible singular  $M$ -matrix. So we have  $B, C \geq 0$ , and  $A$  and  $D$  are nonsingular  $M$ -matrices (see [11], for example). Therefore, the matrix  $I \otimes A + D^T \otimes I$  is also a nonsingular  $M$ -matrix.

By (6) the equation  $-\mathcal{R}'_X(Z) = W$  is equivalent to

$$(I \otimes (A - XC) + (D - CX)^T \otimes I)\text{vec}(Z) = \text{vec}(W),$$

where the  $\text{vec}$  operator stacks the columns of a matrix into one long vector. When the matrix  $I \otimes (A - XC) + (D - CX)^T \otimes I$  is a nonsingular  $M$ -matrix, the map  $\mathcal{R}'_X$  is invertible and we know from Theorem 2 that  $-\mathcal{R}'_X(Z) \geq 0$  implies  $Z \geq 0$  and that  $-\mathcal{R}'_X(Z) > 0$  implies  $Z > 0$ .

We note that  $\mathcal{R}''_X$  in (7) is independent of  $X$  and that  $\mathcal{R}''_X(Z_1, Z_2) \geq 0$  whenever  $Z_1 \geq 0$  and  $Z_2 \geq 0$ , since  $C \geq 0$ . When  $K$  is irreducible, we have  $C \neq 0$  and thus  $\mathcal{R}''_X(Z_1, Z_2) > 0$  whenever  $Z_1 > 0$  and  $Z_2 > 0$ .

The next lemma displays three properties that are preserved in each Newton iteration. Note that these properties are closely related to the properties proved by induction for Newton's method in the proof of Theorem 2.3 in [14]. The main difference is that the first property here is no longer about two consecutive iterates from Newton's method. This change is necessary since we would like to have the freedom to switch to a different method after one or more Newton iterations, or to switch to Newton's method after having used a different method.

**Lemma 2** *Suppose that a matrix  $X$  is such that*

- (i)  $\mathcal{R}(X) \geq 0$ ,
- (ii)  $0 \leq X \leq S$ , and  $0 \leq X < S$  when  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ ,
- (iii)  $I \otimes (A - XC) + (D - CX)^T \otimes I$  is a nonsingular  $M$ -matrix.

*Then the matrix*

$$Y = X - (\mathcal{R}'_X)^{-1} \mathcal{R}(X) \quad (18)$$

*is well defined, and*

- (a)  $\mathcal{R}(Y) \geq 0$ ,
- (b)  $0 \leq Y \leq S$ , and  $0 \leq Y < S$  when  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ ,
- (c)  $I \otimes (A - YC) + (D - CY)^T \otimes I$  is a nonsingular  $M$ -matrix.

*Proof* The matrix  $Y$  is well defined by (iii). By the Taylor formula and (18) we have

$$\mathcal{R}(Y) = \mathcal{R}(X) + \mathcal{R}'_X(Y - X) + \frac{1}{2} \mathcal{R}''_X(Y - X, Y - X) = \frac{1}{2} \mathcal{R}''_X(Y - X, Y - X).$$

So (a) is proved since we have  $Y - X \geq 0$  by (i), (iii) and (18).

To prove (b), note that  $Y \geq X \geq 0$  and

$$\mathcal{R}(S) = \mathcal{R}(X) + \mathcal{R}'_X(S - X) + \frac{1}{2} \mathcal{R}''_X(S - X, S - X). \quad (19)$$

So

$$\begin{aligned} -\mathcal{R}'_X(S - Y) &= \mathcal{R}'_X(Y - X) - \mathcal{R}'_X(S - X) \\ &= -\mathcal{R}(X) - \mathcal{R}'_X(S - X) \\ &= \frac{1}{2} \mathcal{R}''_X(S - X, S - X), \end{aligned}$$

where we have used (19) and  $\mathcal{R}(S) = 0$ . Since  $S - X \geq 0$  by (ii), we have  $-\mathcal{R}'_X(S - Y) \geq 0$  and thus  $S - Y \geq 0$ . If  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ , then we have  $S - X > 0$  by (ii). In this case, we have  $-\mathcal{R}'_X(S - Y) > 0$  and thus  $S - Y > 0$ . This proves (b).

We now prove (c). If  $K$  is a nonsingular  $M$ -matrix or an irreducible singular  $M$ -matrix with  $u_1^T v_1 \neq u_2^T v_2$ , then  $I \otimes (A - SC) + (D - CS)^T \otimes I$  is a nonsingular  $M$ -matrix by Theorem 1. Since  $0 \leq Y \leq S$ , it follows from Lemma 1 that  $I \otimes (A - YC) + (D - CY)^T \otimes I$  is a nonsingular  $M$ -matrix. If  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ , then  $0 \leq Y < S$ . It follows from

$$\mathcal{R}(S) = \mathcal{R}(Y) + \mathcal{R}'_Y(S - Y) + \frac{1}{2} \mathcal{R}''_Y(S - Y, S - Y)$$

that

$$-\mathcal{R}'_Y(S - Y) = \mathcal{R}(Y) + \frac{1}{2} \mathcal{R}''_Y(S - Y, S - Y) > 0.$$

Thus

$$(I \otimes (A - YC) + (D - CY)^T \otimes I) \text{vec}(S - Y) > 0.$$

So  $I \otimes (A - YC) + (D - CY)^T \otimes I$  is a nonsingular  $M$ -matrix by Theorem 2.  $\square$

The next lemma is a generalization of Lemma 2, and will be proved using Lemma 2.

**Lemma 3** *Suppose that a matrix  $X$  is such that*

- (i)  $\mathcal{R}(X) \geq 0$ ,
- (ii)  $0 \leq X \leq S$ , and  $0 \leq X < S$  when  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ ,
- (iii)  $I \otimes (A - XC) + (D - CX)^T \otimes I$  is a nonsingular  $M$ -matrix.

Then for any matrix  $Z$  with  $0 \leq Z \leq X$  the matrix

$$Y = X - (\mathcal{R}'_Z)^{-1} \mathcal{R}(X)$$

is well defined, and

- (a)  $\mathcal{R}(Y) \geq 0$ ,
- (b)  $0 \leq Y \leq S$ , and  $0 \leq Y < S$  when  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ ,
- (c)  $I \otimes (A - YC) + (D - CY)^T \otimes I$  is a nonsingular  $M$ -matrix.

*Proof* Since  $0 \leq Z \leq X$ , we know from Lemma 1 and (iii) that  $I \otimes (A - ZC) + (D - CZ)^T \otimes I$  is also a nonsingular  $M$ -matrix. So the matrix  $Y$  is well defined, and  $Y \geq X \geq 0$ . Let  $\hat{Y} = X - (\mathcal{R}'_X)^{-1} \mathcal{R}(X)$ . It follows from Lemma 1 that  $0 \leq \text{vec}(Y) \leq \text{vec}(\hat{Y})$  and thus  $0 \leq Y \leq \hat{Y}$ . So (b) is true since by Lemma 2 we have  $\hat{Y} \leq S$ , and  $\hat{Y} < S$  when  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ . Also, (c) is true by Lemma 1 since  $I \otimes (A - \hat{Y}C) + (D - C\hat{Y})^T \otimes I$  is a nonsingular  $M$ -matrix by Lemma 2. Finally, (a) is true since we have

$$\begin{aligned} \mathcal{R}(Y) &= \mathcal{R}(X) + \mathcal{R}'_X(Y - X) + \frac{1}{2} \mathcal{R}''_X(Y - X, Y - X) \\ &= \mathcal{R}(X) + \mathcal{R}'_Z(Y - X) + (\mathcal{R}'_X - \mathcal{R}'_Z)(Y - X) + \frac{1}{2} \mathcal{R}''_X(Y - X, Y - X) \\ &= \mathcal{R}''_X(X - Z, Y - X) + \frac{1}{2} \mathcal{R}''_X(Y - X, Y - X), \end{aligned}$$

and  $X - Z \geq 0, Y - X \geq 0$ .  $\square$

From Lemma 3 we can easily get the following monotone convergence result for a modified Newton method.

**Theorem 3** *Let  $S$  be the minimal nonnegative solution of the MARE (1). For any initial guess  $X_0$  satisfying*

- (i)  $\mathcal{R}(X_0) \geq 0$ ,
- (ii)  $0 \leq X_0 \leq S$ , and  $0 \leq X_0 < S$  when  $K$  is an irreducible singular  $M$ -matrix with  $u_1^T v_1 = u_2^T v_2$ ,



(iii)  $I \otimes (A - X_0 C) + (D - C X_0)^T \otimes I$  is a nonsingular  $M$ -matrix,

the modified Newton method

$$X_{i+1} = X_i - (\mathcal{R}'_{X_{k_i}})^{-1} \mathcal{R}(X_i), \quad 0 \leq k_i \leq i,$$

generates a sequence  $\{X_i\}$  with  $X_k \leq X_{k+1} \leq S$  for all  $k \geq 0$ , and  $\lim_{i \rightarrow \infty} X_i = S$ .

*Proof* Note that  $0 \leq X_0 \leq S$  by assumption. Assume that  $0 \leq X_0 \leq X_1 \leq \dots \leq X_i \leq S$  for  $i \geq 0$ . Then  $0 \leq X_{k_j} \leq X_j$  for  $j = 0, 1, \dots, i$ . So Lemma 3 applies when we generate  $X_1, X_2, \dots, X_{i+1}$ . In particular, we have  $\mathcal{R}(X_i) \geq 0$ ,  $I \otimes (A - X_i C) + (D - C X_i)^T \otimes I$  is a nonsingular  $M$ -matrix, and  $X_{i+1} \leq S$ . Since  $0 \leq X_{k_i} \leq X_i$ ,  $I \otimes (A - X_{k_i} C) + (D - C X_{k_i})^T \otimes I$  is also a nonsingular  $M$ -matrix by Lemma 1, and thus  $X_{i+1} \geq X_i$ . Therefore, we have shown that  $0 \leq X_0 \leq X_1 \leq \dots \leq X_{i+1} \leq S$ .

By the induction principle, the sequence  $\{X_i\}$  is well defined, monotonically increasing, and bounded above by  $S$ . So it has a limit  $X_*$ . Letting  $i \rightarrow \infty$  in  $\mathcal{R}(X_i) = -\mathcal{R}'_{X_{k_i}}(X_{i+1} - X_i)$  and noting that  $\{X_{k_i}\}$  is bounded, we see that  $X_*$  is a nonnegative solution of (1). Since  $X_* \leq S$  and  $S$  is the minimal nonnegative solution of (1), we have  $X_* = S$ .  $\square$

Note that the conditions on  $X_0$  in Theorem 3 are always satisfied for  $X_0 = 0$ . If  $K$  is a nonsingular  $M$ -matrix or an irreducible singular  $M$ -matrix with  $u_1^T v_1 \neq u_2^T v_2$ , then each of the (basic) fixed-point iterations in [11], with  $\widehat{X}_0 = 0$ , produces a sequence  $\{\widehat{X}_i\}$  such that  $\widehat{X}_0 \leq \widehat{X}_1 \leq \widehat{X}_2 \leq \dots \leq S$  and  $\{\widehat{X}_i\}$  converges to  $S$  linearly. Since  $I \otimes (A - SC) + (D - CS)^T \otimes I$  is a nonsingular  $M$ -matrix by Theorem 1, it follows from Lemma 1 that  $I \otimes (A - \widehat{X}_i C) + (D - C \widehat{X}_i)^T \otimes I$  is a nonsingular  $M$ -matrix for any  $i \geq 1$ . We can also show that  $\mathcal{R}(\widehat{X}_i) \geq 0$  for each  $i \geq 1$ , as in the proof of [16, Proposition 4.1]. Therefore, for any  $i \geq 1$ , we can use  $\{\widehat{X}_i\}$  as an initial guess  $X_0$  in Theorem 3.

**Corollary 1** *Let  $S$  be the minimal nonnegative solution of the MARE (1). For any initial guess  $X_0$  as in Theorem 3, the sequence  $\{X_k\}$  generated by the Newton–Shamanskii method (which includes Newton’s method, Chebychev’s method and modified Chebyshev method) is monotonically increasing and converges to  $S$ .*

*Proof* The result is true since the sequence here is a subsequence of the sequence generated by a proper modified Newton method in Theorem 3.  $\square$

We note that the above result holds even when the integer  $r$  is dependent on  $i$  in (12) of Algorithm 1.

Each Newton-like method has the form  $X_{k+1} = X_k + \Delta_k$ , and we have established monotone convergence by proving  $\Delta_k \geq 0$ . In actual computations it is possible to have some tiny negative entries in the computed  $\Delta_k$ . A simple strategy to deal with this situation is to set these tiny negative entries to 0.

## 4 Numerical results

In this section we will identify two situations where Newton-like methods can be more efficient than the doubling algorithm. Our numerical results also show that Chebyshev's method and modified Chebyshev method have better efficiency than Newton's method in actual computations, as suggested by the efficiency indices.

There have been several different versions of the doubling algorithm for the MARE (1). First, a structure-preserving doubling algorithm (SDA) is introduced in [18] and analyzed for the case where  $K$  is a nonsingular  $M$ -matrix. The same algorithm is then analyzed in [15] for the case where  $K$  is an irreducible singular  $M$ -matrix. The algorithm uses the Cayley transform with a parameter  $\gamma \geq \max_{i,j} \{a_{ii}, d_{jj}\}$ , where  $a_{ii}$  and  $d_{jj}$  are the diagonal entries of  $A$  and  $D$ , respectively. It produces four sequences  $\{E_k\}, \{F_k\}, \{G_k\}, \{H_k\}$ , and one has

$$\limsup_{k \rightarrow \infty} \sqrt[2k]{\|H_k - S\|} \leq \rho((V + \gamma I)^{-1}(V - \gamma I)) \cdot \rho((W + \gamma I)^{-1}(W - \gamma I)), \quad (20)$$

where  $V = D - CS$  and  $W = A - B\widehat{S}$ . Note that the eigenvalues of  $A - B\widehat{S}$  are the same as those of  $A - SC$  [11, 12]. The upper bound in (20) is usually tight. It is strictly less than 1 in the non-critical case and is exactly 1 in the critical case. The convergence of SDA in the critical case has been shown to be linear in [8]. It is shown in [15] that the upper bound in (20) is minimized when  $\gamma = \max_{i,j} \{a_{ii}, d_{jj}\}$ .

Later, the algorithm SDA-ss (SDA combined with the shrink-and-shift technique) is proposed in [6]. The algorithm is significantly faster than SDA in some situations.

Recently, Wang, Wang, and Li [29] present an alternating-directional doubling algorithm (ADDA) for the MARE (1). The algorithm uses the generalized Cayley transform with two parameters  $\alpha \geq \max_i a_{ii}$  and  $\beta \geq \max_j d_{jj}$ . It generates four new sequences  $\{E_k\}, \{F_k\}, \{G_k\}, \{H_k\}$ , and one now has

$$\limsup_{k \rightarrow \infty} \sqrt[2k]{\|H_k - S\|} \leq \rho((V + \alpha I)^{-1}(V - \beta I)) \cdot \rho((W + \beta I)^{-1}(W - \alpha I)), \quad (21)$$

where the upper bound in (21) is usually tight. It is shown in [29] that this upper bound is minimized when  $\alpha = \max_i a_{ii}$  and  $\beta = \max_j d_{jj}$  and that ADDA is faster than SDA of [18] and SDA-ss of [6]. In particular, when  $\max_i a_{ii}$  and  $\max_j d_{jj}$  differ significantly, ADDA with optimal  $\alpha$  and  $\beta$  can be significantly faster than SDA with optimal  $\gamma$  [29]. When  $\alpha = \beta$ , however, ADDA reduces to SDA.

We can now identify one situation where Newton-like methods can be more efficient than the doubling algorithms. This is the situation where  $\max_i a_{ii} \gg \min_i a_{ii}$  and  $\max_j d_{jj} \gg \min_j d_{jj}$ , but  $\max_i a_{ii} \approx \max_j d_{jj}$ .

To illustrate this, we have performed some experiments in MATLAB. We use the normalized residual

$$\text{NRes} = \frac{\|XCX - XD - AX + B\|_1}{\|X\|_1(\|C\|_1\|X\|_1 + \|A\|_1 + \|D\|_1) + \|B\|_1}$$

to measure the accuracy of an approximate solution  $X$  for the MARE (1). Our stopping criterion is  $\text{NRes} < 10^{-14}$ . Since  $\max a_{ii}$  and  $\max d_{jj}$  do not differ by too much in our examples, SDA, SDA-ss and ADDA have roughly the same performance. Thus we only need to compare Newton-like methods with SDA of [18], which will be referred to as the doubling algorithm. We use  $X_0 = 0$  for all Newton-like methods, unless otherwise specified. For  $m = n$  sufficiently large, Newton's method requires  $68n^3$  flops each iteration, Chebyshev's method requires  $82n^3$  flops each iteration, and modified Chebyshev method requires  $98n^3$  flops each iteration. The doubling algorithm requires  $\frac{64}{3}n^3$  flops each iteration (Checking the stopping criterion using the residual would require an additional  $6n^3$  flops each iteration for the doubling algorithm. But in practice we can use a different stopping criterion with a negligible cost).

*Example 1* We consider the MARE (1) with

$$A = \begin{bmatrix} 3+p & -1 & -p & 0 \\ 0 & 3 & -1 & \\ -2 & 0 & 3 & \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}, \quad D = \begin{bmatrix} 3+p & -1 & -p & 0 \\ 0 & 3 & -1 & \\ -1 & 0 & 3 & \end{bmatrix},$$

where  $p$  is a parameter. Then matrix  $K$  in (2) is an irreducible singular  $M$ -matrix for every  $p \geq 0$ .

In Table 1 we report the number of iterations and the normalized residual for Newton's method, Chebyshev's method, the modified Chebyshev method, and the doubling algorithm for different values of  $p$ . The case  $p = 0$  is the perfect situation for the doubling algorithm, since all diagonal entries of  $A$  and  $D$  are equal. In this case, the doubling algorithm requires 7 iterations, the same as Newton's method. As  $p$  gets larger, the number of iterations for the Newton-like methods does not increase at all. For all the  $p$  values in the table we have the non-critical case, so Newton's method converges quadratically, while Chebyshev's method has third order convergence and modified Chebyshev method has fourth order convergence. The doubling algorithm converges quadratically for each  $p$  value, but requires more iterations as  $p$  gets larger. This is easily explained by (20) since one of the spectral radius in (20) is exactly 1 for any  $p \geq 0$  and the other is very close to 1 when  $\gamma = 3 + p$  is very large.

To illustrate the comparison of these methods for the MARE (1) with larger coefficient matrices, we consider the MARE (1) with  $m = n$  and with the matrix  $K$  in (2) given as follows.

**Table 1** Iteration count and normalized residual for Example 1

$p$	Newton	Chebyshev	Modified Chebyshev	Doubling
0	7 $2.6 \times 10^{-15}$	5 $3.0 \times 10^{-17}$	4 $4.0 \times 10^{-17}$	7 $5.3 \times 10^{-17}$
$10^2$	7 $1.7 \times 10^{-17}$	5 $1.7 \times 10^{-17}$	4 $1.0 \times 10^{-17}$	12 $8.6 \times 10^{-17}$
$10^4$	6 $8.9 \times 10^{-15}$	5 $1.5 \times 10^{-17}$	4 $2.1 \times 10^{-17}$	18 $6.8 \times 10^{-17}$
$10^6$	6 $1.0 \times 10^{-16}$	4 $8.5 \times 10^{-16}$	4 $1.5 \times 10^{-17}$	24 $2.4 \times 10^{-16}$
$10^8$	6 $2.2 \times 10^{-17}$	4 $2.1 \times 10^{-17}$	3 $2.2 \times 10^{-15}$	30 $3.7 \times 10^{-15}$

*Example 2* We generate the matrix  $K$  using MATLAB:

```
n=100; p=10^6;
K=rand(2*n); K=diag(K*ones(2*n,1))-K;
K(1,1)=K(1,1)+p; K(1,2)=K(1,2)-p;
K(n+1,n+1)=K(n+1,n+1)+p; K(n+1,n+2)=K(n+1,n+2)-p;
```

The matrix  $K$  is again an irreducible singular  $M$ -matrix. The results for the corresponding MARE (1) are reported in Table 2. In terms of flop counts, the modified Chebyshev method and Chebyshev's method are more efficient than Newton's method for this example. The same can be said even when the Lyapunov equation is solved by the Hessenberg–Schur method, but the advantage is very small for this example. In terms of flop counts, all three Newton-like methods have roughly the same efficiency as the doubling algorithm if the Lyapunov equation is solved by the Hessenberg–Schur method. When the Lyapunov equation is solved by the Bartels–Stewart algorithm, the flop count for the modified Chebyshev method is still slightly larger than that for the doubling algorithm, but the Newton-like methods have provided smaller normalized residuals. Using a larger value of  $p$  would make Newton-like methods more favorable. We also find that the Newton-like methods always produce smaller normalized residuals than the doubling algorithm if we do not apply the stopping criterion. This is not surprising since Newton-like methods are correction methods and the doubling algorithm is not.

**Table 2** Iteration count and normalized residual for Example 2

Newton	Chebyshev	Modified Chebyshev	Doubling
11 $1.2 \times 10^{-16}$	8 $1.0 \times 10^{-18}$	6 $1.0 \times 10^{-17}$	25 $1.9 \times 10^{-15}$

We now identify a situation where Newton-like methods can be significantly more efficient than the doubling algorithm. Suppose we need to solve the MARE (1) involving a parameter. Such a situation could happen when there is some uncertainty for an entry of the matrix  $K$  in (2) and we would like to determine numerically how sensitive the minimal solution is to the changes in this entry. We use the following example to illustrate this and we note that the MARE (1) involving multiple parameters can be considered in the same way.

*Example 3* We consider the MARE (1) with  $m = n = 100$  and

$$A = \begin{bmatrix} 2+p & -1 & -p & & & \\ & & 3 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & 3 & -1 \\ -1 & & & & & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & & 1 & 1 & \\ & & & & q & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ & & & & & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 3+p & -1 & -p & & & \\ & & 3 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & 3 & -1 \\ -1 & & & & & 2 \end{bmatrix},$$

where  $p, q \geq 0$  are parameters. Then the matrix  $K$  is irreducible. It is a singular  $M$ -matrix for  $q = 2$  and a nonsingular  $M$ -matrix for  $0 \leq q < 2$ .

We first take  $p = 0$  and try to compute the minimal solution of the MARE for some values of  $q$  near 2, say  $q = 1.98 + 0.0001j$ ,  $j = 0, 1, \dots, 200$ . With the stopping criterion  $\text{NRes} < 10^{-14}$ , Newton's method, Chebyshev's method, the modified Chebyshev method, and the doubling algorithm take 11, 8, 6, and 11 iterations, respectively, for  $j = 0$ . If we use the doubling algorithm to compute the minimal solution  $S^{(j)}$  for  $j = 0, 1, \dots, 200$ , then the algorithm will need 11 or 12 iterations for each  $j$ . We cannot use the computed solution for  $j = 0$  to reduce the number of iteration for computing the minimal solution for  $j = 1$ , for example, since the doubling algorithm is not a correction method. If we use Newton's method, Chebyshev's method or the modified Chebyshev method, then we can use the approximate solution  $X^{(j)}$  for  $S^{(j)}$  as the initial guess for  $S^{(j+1)}$ . Since  $S^{(j)} \leq S^{(j+1)}$  by [11, Theorem 2.4], monotone convergence is still guaranteed (in exact arithmetic) by Theorem 3 and Lemma 3 when this cost-saving strategy is used. If we use Newton's method, then we need 2 iterations for  $1 \leq j \leq 162$  and 3 iterations for  $163 \leq j \leq 200$ . If we use Chebyshev's method, then we need 2 iterations for  $1 \leq j \leq 200$ . If we use the modified Chebyshev method, then we need 1 iteration for  $1 \leq j \leq 125$  and 2 iterations for  $126 \leq j \leq 200$ . We also have complete freedom to switch between the three methods. For example, we can use the modified Chebyshev method for  $1 \leq j \leq 126$ . After noticing that the method begins to requires 2 iterations for  $j = 126$ , we use the less expensive Newton's method instead.

We then find that Newton's method requires 2 iterations for  $127 \leq j \leq 162$  and starts to require 3 iterations for  $j = 163$ . We can then use Chebyshev's method for  $164 \leq j \leq 200$  with 2 iterations for each  $j$ .

To make the Newton-like methods even more favorable, we take  $p = 10^5$  and compute the minimal solution of the MARE for  $q = 1.98 + 0.0001j$ ,  $j = 0, 1, \dots, 200$ , again with the stopping criterion  $\text{NRes} < 10^{-14}$ . For  $j = 0$ , Newton's method, Chebyshev's method, the modified Chebyshev method, and the doubling algorithm take 10, 7, 6, and 24 iterations, respectively. If we use the doubling algorithm to compute the minimal solution  $S^{(j)}$  for  $j = 0, 1, \dots, 200$ , then the algorithm will need 24 or 25 iterations for each  $j$ . If we use Newton's method, Chebyshev's method or the modified Chebyshev method, then we can use the same cost-saving strategy as for the case  $p = 0$ . If we use Newton's method, then we need 2 iterations for  $1 \leq j \leq 200$ . If we use Chebyshev's method or modified Chebyshev's method, then we need just 1 iteration for  $1 \leq j \leq 200$ .

## 5 Conclusions

We have proved monotone convergence results for some Newton-like methods for  $M$ -matrix algebraic Riccati equations. We have explained that the Chebyshev method of order three and a two-step modified Chebyshev method of order four can be more efficient than Newton's method. We have also pointed out that Newton-like methods can be more efficient than the doubling algorithm in some situations.

## References

1. Bai, Z.-Z., Guo, X.-X., Xu, S.-F.: Alternately linearized implicit iteration methods for the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations. *Numer. Linear Algebra Appl.* **13**, 655–674 (2006)
2. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation  $AX + XB = C$ . *Comm. ACM* **15**, 820–826 (1972)
3. Berman, A., Plemmons, R.J.: *Nonnegative Matrices in the Mathematical Sciences*, revised reprint of the 1979 Academic Press original. SIAM, Philadelphia (1994)
4. Bini, D.A., Iannazzo, B., Latouche, G., Meini, B.: On the solution of Riccati equations arising in fluid queues. *Linear Algebra Appl.* **413**, 474–494 (2006)
5. Bini, D.A., Iannazzo, B., Meini, B.: *Numerical Solution of Algebraic Riccati Equations*. SIAM, Philadelphia (2012)
6. Bini, D.A., Meini, B., Poloni, F.: Transforming algebraic Riccati equations into unilateral quadratic matrix equations. *Numer. Math.* **116**, 553–578 (2010)
7. Candela, V., Marquina, A.: Recurrence relations for rational cubic methods II: The Chebyshev method. *Computing* **45**, 355–367 (1990)
8. Chiang, C.-Y., Chu, E.K.-W., Guo, C.-H., Huang, T.-M., Lin, W.-W., Xu, S.-F.: Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. *SIAM J. Matrix Anal. Appl.* **31**, 227–247 (2009)
9. Gao, Y.-H., Bai, Z.-Z.: On inexact Newton methods based on doubling iteration scheme for non-symmetric algebraic Riccati equations. *Numer. Linear Algebra Appl.* **18**, 325–341 (2011)

10. Golub, G.H., Nash, S., Van Loan, C.: A Hessenberg–Schur method for the problem  $AX + XB = C$ . *IEEE Trans. Autom. Control.* **24**, 909–913 (1979)
11. Guo, C.-H.: Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for  $M$ -matrices. *SIAM J. Matrix Anal. Appl.* **23**, 225–242 (2001)
12. Guo, C.-H.: A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation. *Linear Algebra Appl.* **357**, 299–302 (2002)
13. Guo, C.-H.: Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models. *J. Comput. Appl. Math.* **192**, 353–373 (2006)
14. Guo, C.-H., Higham, N.J.: Iterative solution of a nonsymmetric algebraic Riccati equation. *SIAM J. Matrix Anal. Appl.* **29**, 396–412 (2007)
15. Guo, C.-H., Iannazzo, B., Meini, B.: On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation. *SIAM J. Matrix Anal. Appl.* **29**, 1083–1100 (2007)
16. Guo, C.-H., Laub, A.J.: On the iterative solution of a class of nonsymmetric algebraic Riccati equations. *SIAM J. Matrix Anal. Appl.* **22**, 376–391 (2000)
17. Guo, X.-X., Bai, Z.-Z.: On the minimal nonnegative solution of nonsymmetric algebraic Riccati equation. *J. Comput. Math.* **23**, 305–320 (2005)
18. Guo, X.-X., Lin, W.-W., Xu, S.-F.: A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation. *Numer. Math.* **103**, 393–412 (2006)
19. Higham, N.J.: *Functions of Matrices: Theory and Computation*. SIAM, Philadelphia (2008)
20. Iannazzo, B., Poloni, F.: A subspace shift technique for nonsymmetric algebraic Riccati equations associated with an  $M$ -matrix. *Numer. Linear Algebra Appl.* (2012). doi:10.1002/nla.1836
21. Juang, J.: Existence of algebraic matrix Riccati equations arising in transport theory. *Linear Algebra Appl.* **230**, 89–100 (1995)
22. Juang, J., Lin, W.-W.: Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices. *SIAM J. Matrix Anal. Appl.* **20**, 228–243 (1998)
23. Kelley, C.T.: A Shamanskii-like acceleration scheme for nonlinear equations at singular roots. *Math. Comput.* **47**, 609–623 (1986)
24. Lin, Y., Bao, L.: Convergence analysis of the Newton–Shamanskii method for a nonsymmetric algebraic Riccati equation. *Numer. Linear Algebra Appl.* **15**, 535–546 (2008)
25. Rogers, L.C.G.: Fluid models in queueing theory and Wiener–Hopf factorization of Markov chains. *Ann. Appl. Probab.* **4**, 390–413 (1994)
26. Sargolzaei, P., Soleymani, F.: Accurate fourteenth-order methods for solving nonlinear equations. *Numer. Algorithms* **58**, 513–527 (2011)
27. Sharma, J.R., Sharma, R.: A new family of modified Ostrowski’s methods with accelerated eighth order convergence. *Numer. Algorithms* **54**, 445–458 (2010)
28. Traub, J.F.: *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs (1964)
29. Wang, W.-G., Wang, W.-C., Li, R.-C.: Alternating-directional doubling algorithm for  $M$ -matrix algebraic Riccati equations. *SIAM J. Matrix Anal. Appl.* **33**, 170–194 (2012)
30. Xue, J., Xu, S., Li, R.-C.: Accurate solutions of  $M$ -matrix algebraic Riccati equations. *Numer. Math.* **120**, 671–700 (2012)