Computers and Mathematics with Applications 62 (2011) 4178–4184

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

On the convergence rate of an iterative method for solving nonsymmetric algebraic Riccati equations

Yiqin Lin [a], Liang Bao [b,*], Qinghua Wu [a]

[a] Department of Mathematics and Computational Science, Hunan University of Science and Engineering, Yongzhou 425100, PR China

[b] Department of Mathematics, East China University of Science and Technology, Shanghai 200237, PR China

ARTICLE INFO

Article history:
Received 18 April 2011
Received in revised form 15 September 2011
Accepted 29 September 2011

Keywords:
Nonsymmetric algebraic Riccati equation
Transport theory
Minimal positive solution
Iterative method
Convergence rate

ABSTRACT

This paper is devoted to the convergence analysis of an iterative method for solving a nonsymmetric algebraic Riccati equation arising in transport theory. We give the convergence rate, and show that the iterative method converges linearly in one case and sublinearly in the other case.

1. Introduction

The nonsymmetric algebraic Riccati equation arising in transport theory is

$$XCX - XD - AX + B = 0, \tag{1}$$

where $A, B, C, D \in \mathbb{R}^{n\times n}$ are given by

$$A = \Delta - eq^T, \qquad B = ee^T, \qquad C = qq^T, \qquad D = \Gamma - qe^T.$$

Here $e = [1, 1, \ldots, 1]^T$, $q = [q_1, q_2, \ldots, q_n]^T$ with $q_i = \frac{c_i}{2\omega_i}$,

$$\begin{cases} \Delta = \operatorname{diag}([\delta_1, \delta_2, \ldots, \delta_n]) & \text{with } \delta_i = \dfrac{1}{c\omega_i(1+\alpha)}, \\ \Gamma = \operatorname{diag}([\gamma_1, \gamma_2, \ldots, \gamma_n]) & \text{with } \gamma_i = \dfrac{1}{c\omega_i(1-\alpha)}, \end{cases}$$

and $0 < c \le 1$, $0 \le \alpha < 1$, $0 < \omega_n < \cdots < \omega_2 < \omega_1 < 1$, $\sum_{i=1}^n c_i = 1$, $c_i > 0$, $i = 1, 2, \ldots, n$.

Let $P = [P_{ij}] = [q_j/(\delta_i + \gamma_j)]$, $Q = [Q_{ij}] = [q_j/(\delta_j + \gamma_i)]$, and $T = [t_{i,j}] = [1/(\delta_i + \gamma_j)]$. It has been shown in [1–3] that (1) has positive solutions (in the componentwise sense), and the solutions must be of the form

$$X = T \circ (uv^T) = (uv^T) \circ T,$$

* Corresponding author. Tel.: +86 21 64252839; fax: +86 21 64252839.
E-mail addresses: yqlin@yahoo.cn (Y. Lin), nlbao@yahoo.cn (L. Bao), jackwqh@163.com (Q. Wu).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.09.068

where u and v satisfy the vector equation

$$\begin{cases} u = u \circ (Pv) + e, \\ v = v \circ (Qu) + e. \end{cases} \tag{2}$$

The solution of practical interest of (1) is the minimal positive solution, which can be obtained via computing the minimal positive solution of the vector equation (2). Several iterative methods have been proposed in literature for computing the minimal positive solution (u_*, v_*) of (2). Lu [3] developed a simple iterative method to solve (2). A modified simple iterative method was proposed in [4]. A nonlinear block Jacobi method (NBJ) and a nonlinear block Gauss–Seidel method (NBGS) were proposed in [5]. Wu and Huang [6] established two-step relaxation Newton method (TSRN). The iteration sequences generated by these methods mentioned above are all strictly and monotonically increasing, and converge to the minimal positive solution (u_*, v_*). Guo and Lin [7] analyzed the convergence rates of these iterative methods, and showed that all of these iterative methods converge linearly when $(\alpha, c) \neq (0, 1)$, and however sublinearly when $(\alpha, c) = (0, 1)$. In [8–10], three quadratically convergent iterative methods were designed. As pointed out in [7], these three methods are more appropriate for the case where (α, c) is relatively close to $(0, 1)$.

Recently, Lin [11] proposed a class of iterative methods for obtaining the minimal positive solution (u_*, v_*) of (2). Let $w = [u^T, v^T]^T$ and

$$F(w) = \begin{bmatrix} u - u \circ (Pv) - e \\ v - v \circ (Qu) - e \end{bmatrix}.$$

The basic iterative scheme in [11] is

$$w_{k+1} = w_k - T_k^{-1} F(w_k), \quad k = 0, 1, 2, \ldots, \tag{3}$$

where $w_k = [u_k^T, v_k^T]^T$ with $w_0 = 0$, and T_k is an approximation to $F'(w_k)$. Here, $F'(w_k)$ denotes the Jacobian of $F(w)$ at w_k. It has been shown that the vector sequence generated by (3) with

$$T_k = \begin{bmatrix} I - \operatorname{diag}(Pv_k) & -\operatorname{diag}(u_k)P \\ 0 & I - \operatorname{diag}(Qu_k) \end{bmatrix} \tag{4}$$

is all strictly and monotonically increasing, and converges to the minimal positive solution $w_* = [u_*^T, v_*^T]^T$.

In this paper, we will prove that the iterative method (3) with T_k given by (4) has the same asymptotic convergence rate as the nonlinear block Gauss–Seidel method in [5], i.e., it converges linearly when $(\alpha, c) \neq (0, 1)$ and sublinearly when $(\alpha, c) = (0, 1)$.

Throughout the paper, we use the following notation. For any matrices $A = [a_{ij}], B = [b_{ij}] \in \mathbb{R}^{m\times n}$, we write $A \geq B$ ($A > B$) if $a_{i,j} \geq b_{ij}$ ($a_{i,j} > b_{ij}$) holds for all i, j. The Hadamard product of A and B is defined by $A \circ B = [a_{ij} \cdot b_{ij}]$. I denotes the identity matrix and 0 denotes the zero vector or zero matrix. The dimensions of these vectors and matrices are conformed with dimensions used in the context. The superscript T denotes the transpose of a vector or a matrix. We denote any consistent norm by $\|\cdot\|$ for a vector or a matrix.

2. Analysis of the convergence rate

It is easy to verify that the iterative scheme (3) with T_k given by (4) is equivalent to

$$u_{k+1} = e - u_k \circ (Pv_k) + u_{k+1} \circ Pv_k + u_k \circ Pv_{k+1}, \tag{5}$$

$$v_{k+1} = v_{k+1} \circ Qu_k + e. \tag{6}$$

It follows from (6) and the second equations of (2) that

$$v_* - v_{k+1} = v_* - e - v_{k+1} \circ Qu_k = v_* \circ (Qu_*) - v_{k+1} \circ Qu_k. \tag{7}$$

From (7) and

$$\begin{aligned} v_* \circ (Qu_*) - v_{k+1} \circ Qu_k - v_* \circ v_{k+1} \circ Q(u_* - u_k) &= (e - v_{k+1}) \circ v_* \circ (Qu_*) + (v_* - e) \circ v_{k+1} \circ Qu_k \\ &= (e - v_{k+1}) \circ (v_* - e) + (v_* - e) \circ (v_{k+1} - e) \\ &= 0, \end{aligned}$$

we obtain

$$v_* - v_{k+1} = v_* \circ v_{k+1} \circ Q(u_* - u_k). \tag{8}$$

By using (5) and the first equations of (2), we have

$$\begin{aligned}
u_* - u_{k+1} &= u_* - e + u_k \circ (Pv_k) - u_{k+1} \circ Pv_k - u_k \circ Pv_{k+1} \\
&= u_* \circ (Pv_*) + u_k \circ (Pv_k) - u_{k+1} \circ Pv_k - u_k \circ Pv_{k+1} \\
&= u_* \circ (Pv_*) - u_* \circ (Pv_k) + u_* \circ (Pv_k) - u_{k+1} \circ Pv_k \\
&\quad + u_k \circ (Pv_k) - u_k \circ (Pv_*) + u_k \circ (Pv_*) - u_k \circ Pv_{k+1} \\
&= u_* \circ P(v_* - v_k) + (Pv_k) \circ (u_* - u_{k+1}) - u_k \circ P(v_* - v_k) + u_k \circ P(v_* - v_{k+1}) \\
&= (u_* - u_k) \circ P(v_* - v_k) + (Pv_k) \circ (u_* - u_{k+1}) + u_k \circ P(v_* \circ v_{k+1} \circ Q(u_* - u_k)),
\end{aligned}$$

which shows

$$(e - Pv_k) \circ (u_* - u_{k+1}) = P(v_* - v_k) \circ (u_* - u_k) + u_k \circ P(v_* \circ v_{k+1} \circ Q(u_* - u_k))$$

or

$$u_* - u_{k+1} = (\mathrm{diag}(e - Pv_k))^{-1}(\mathrm{diag}(P(v_* - v_k)) + \mathrm{diag}(u_k)P\mathrm{diag}(v_* \circ v_{k+1})Q)(u_* - u_k). \tag{9}$$

Let

$$d_k = \begin{bmatrix} u_* - u_k \\ v_* - v_k \end{bmatrix}.$$

From (8) and (9), it follows that

$$d_{k+1} = L_k d_k, \quad k = 0, 1, 2, \ldots,$$

where

$$L_k = \begin{bmatrix} (\mathrm{diag}(e - Pv_k))^{-1}(\mathrm{diag}(P(v_* - v_k)) + \mathrm{diag}(u_k)P\mathrm{diag}(v_* \circ v_{k+1})Q) & 0 \\ \mathrm{diag}(v_* \circ v_{k+1})Q & 0 \end{bmatrix}.$$

By the first equations of (2), we get

$$(e - Pv_*) \circ u_* = e,$$

i.e.,

$$(\mathrm{diag}(e - Pv_*))^{-1} = \mathrm{diag}(u_*).$$

Since $\lim_{k\to\infty} u_k = u_*$ and $\lim_{k\to\infty} v_k = v_*$, we have

$$\begin{aligned}
\lim_{k\to\infty} L_k &= \begin{bmatrix} (\mathrm{diag}(e - Pv_*))^{-1}\mathrm{diag}(u_*)P(\mathrm{diag}(v_* \circ v_*)Q) & 0 \\ \mathrm{diag}(v_* \circ v_*)Q & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{diag}(u_* \circ u_*)P(\mathrm{diag}(v_* \circ v_*)Q) & 0 \\ \mathrm{diag}(v_* \circ v_*)Q & 0 \end{bmatrix} \equiv L(w_*).
\end{aligned}$$

Define

$$\widetilde{L}_k = \begin{bmatrix} (\mathrm{diag}(e - Pv_k))^{-1}\mathrm{diag}(u_k)P\mathrm{diag}(v_* \circ v_{k+1})Q & 0 \\ \mathrm{diag}(v_* \circ v_{k+1})Q & 0 \end{bmatrix},$$

where u_k and v_k are generated by the iterative method (3) with T_k given by (4).

Since $0 = u_0 < u_1 < u_2 < \cdots < u_k < u_{k+1} < \cdots$ and $0 = v_0 < v_1 < v_2 < \cdots < v_k < v_{k+1} < \cdots$, we have

$$0 \le \widetilde{L}_0 \le \widetilde{L}_1 \le \widetilde{L}_2 \le \cdots \le \widetilde{L}_k \le \widetilde{L}_{k+1} \le \cdots.$$

Moreover, we have

$$\lim_{k\to\infty} \widetilde{L}_k = \begin{bmatrix} (\mathrm{diag}(u_* \circ u_*)P\mathrm{diag}(v_* \circ v_*)Q) & 0 \\ \mathrm{diag}(v_* \circ v_*)Q & 0 \end{bmatrix} = L(w_*).$$

Define

$$\tilde{d}_{k+1} = \widetilde{L}_k \tilde{d}_k, \qquad \tilde{d}_0 = d_0.$$

By Theorem 4 in [7], we obtain

$$\limsup_{k\to\infty} \sqrt[k]{\|\tilde{d}_k\|} = \rho(L(w_*)),$$

where $\|\cdot\|$ is any matrix norm and $\rho(\cdot)$ denotes the spectral radius.

Since

$$L_k - \widetilde{L}_k = \begin{bmatrix} (\mathrm{diag}(e - Pv_k))^{-1}\mathrm{diag}(P(v_* - v_k)) & 0 \\ 0 & 0 \end{bmatrix} \geq 0,$$

it is easy to obtain by induction

$$d_k \geq \tilde{d}_k \geq 0 \quad \text{for } k \geq 0.$$

Thus, we have

$$\rho(L(w_*)) = \limsup_{k\to\infty} \sqrt[k]{\|\tilde{d}_k\|} \leq \limsup_{k\to\infty} \sqrt[k]{\|d_k\|}.$$

From $d_k = L_{k-1}L_{k-2}\cdots L_1L_0d_0$, it follows that for any matrix norm $\|\cdot\|$,

$$\|d_k\| \leq \|L_{k-1}\|\,\|L_{k-2}\|\cdots\|L_1\|\,\|L_0\|\,\|d_0\|.$$

Hence,

$$\begin{aligned} \limsup_{k\to\infty} \sqrt[k]{\|d_k\|} &\leq \limsup_{k\to\infty} \sqrt[k]{\|L_{k-1}\|\,\|L_{k-2}\|\cdots\|L_1\|\,\|L_0\|\,\|d_0\|} \\ &= \limsup_{k\to\infty} \sqrt[k]{\|L_{k-1}\|\,\|L_{k-2}\|\cdots\|L_1\|\,\|L_0\|}. \end{aligned}$$

Since $\lim_{k\to\infty} L_k = L(w_*)$, we obtain

$$\limsup_{k\to\infty} \sqrt[k]{\|L_{k-1}\|\,\|L_{k-2}\|\cdots\|L_1\|\,\|L_0\|} = \|L(w_*)\|.$$

Therefore, for any matrix norm $\|\cdot\|$,

$$\limsup_{k\to\infty} \sqrt[k]{\|d_k\|} \leq \|L(w_*)\|,$$

which shows that

$$\limsup_{k\to\infty} \sqrt[k]{\|d_k\|} \leq \rho(L(w_*)).$$

In summary, we obtain the following result on the asymptotic convergence rate of the iterative method (3) with T_k given by (4).

Theorem 1. *For the iterative method (3) with T_k given by (4) and $w_0 = 0$, we have*

$$\limsup_{k\to\infty} \sqrt[k]{\|d_k\|} = \rho(L(w_*)).$$

The following theorem shows that the asymptotic convergence rate of the iterative method is linear when $(\alpha, c) \neq (0, 1)$ and sublinear when $(\alpha, c) = (0, 1)$.

Theorem 2. *If $(\alpha, c) = (0, 1)$, then $\rho(L(w_*)) = 1$. If $(\alpha, c) \neq (0, 1)$, then $\rho(L(w_*)) < 1$.*

Proof. Define

$$G(w_*) = \begin{bmatrix} 0 & \mathrm{diag}(u_* \circ u_*)P \\ 0 & \mathrm{diag}(v_* \circ v_*)Q\mathrm{diag}(u_* \circ u_*)P \end{bmatrix}.$$

It has been shown in [7, Theorem 9] that $\rho(G(w_*)) = 1$ when $(\alpha, c) = (0, 1)$, and $\rho(G(w_*)) < 1$ when $(\alpha, c) \neq (0, 1)$. Then, this theorem follows from

$$\begin{aligned} \rho(G(w_*)) &= \rho(\mathrm{diag}(v_* \circ v_*)Q\,\mathrm{diag}(u_* \circ u_*)P) \\ &= \rho(\mathrm{diag}(u_* \circ u_*)P\mathrm{diag}(v_* \circ v_*)Q) = \rho(L(w_*)). \quad \square \end{aligned}$$

It has been shown in [7] that the NBGS proposed in [5] has the asymptotic convergence rate $\rho(G(w_*))$. Thus, the iterative method (3) with T_k given by (4) has the same asymptotic convergence rate as the nonlinear block Gauss–Seidel method.

The iterative method (3) with

$$T_k = \begin{bmatrix} I - \mathrm{diag}(Pv_k) & 0 \\ 0 & I - \mathrm{diag}(Qu_k) \end{bmatrix} \tag{10}$$

is also considered in [11]. It is equivalent to

$$\begin{cases} u_{k+1} = (I - \mathrm{diag}(Pv_k))^{-1}e, \\ v_{k+1} = (I - \mathrm{diag}(Qu_k))^{-1}e, \end{cases} \tag{11}$$

and therefore is the same as the NBJ proposed in [5].

The iterative method (3) with

$$T_k = \begin{bmatrix} I - \mathrm{diag}(Pv_k) & 0 \\ \mathrm{diag}(v_k)Q & I - \mathrm{diag}(Qu_k) \end{bmatrix}$$

is also mentioned in [11]. Following similar arguments as above, we can show that this method has the same asymptotic convergence rate as the iterative method (3) with T_k given by (4), and therefore converges linearly when $(\alpha, c) \neq (0, 1)$ and sublinearly when $(\alpha, c) = (0, 1)$.

3. Comparison with NBJ, NBGS and TSRN

In this section, we will compare the method considered in this paper with NBJ, NBGS and TSRN on the computation complexity and the parallel potential.

The iterative scheme (3) with T_k given by (4) can be formulated as

$$\begin{cases} u_{k+1} = (I - \mathrm{diag}(Pv_k))^{-1}e + (I - \mathrm{diag}(Pv_k))^{-1}\mathrm{diag}(u_k)P\left((I - \mathrm{diag}(Qu_k))^{-1}e - v_k\right), \\ v_{k+1} = (I - \mathrm{diag}(Qu_{k+1}))^{-1}e. \end{cases} \tag{12}$$

The computational scheme for NBJ is given by (11), while the iterative scheme of NBGS is

$$\begin{cases} u_{k+1} = (I - \mathrm{diag}(Pv_k))^{-1}e, \\ v_{k+1} = (I - \mathrm{diag}(Qu_{k+1}))^{-1}e. \end{cases} \tag{13}$$

Let Φ and Ψ be diagonal matrices, whose diagonal elements are defined by

$$\Phi_{ii} = \begin{cases} P_{ii}, & \text{if } i \text{ is odd}, \\ 0, & \text{if } i \text{ is even}, \end{cases} \qquad \Psi_{ii} = \begin{cases} 0, & \text{if } i \text{ is odd}, \\ Q_{ii}, & \text{if } i \text{ is even}. \end{cases}$$

The iteration for TSRN was given in an elementwise fashion in [6]. It can be formulated as the following vector form

$$\begin{cases} u_{k+1/2} = (I - \mathrm{diag}(Pv_k))^{-1}e, \\ v_{k+1/2} = (I - \mathrm{diag}(Qu_k))^{-1}e, \\ u_{k+1} = (I - \mathrm{diag}(Pv_{k+1/2}))^{-1}(e - (\Phi u_{k+1/2}) \circ v_{k+1/2} + (I - \mathrm{diag}(Qu_{k+1/2}))^{-1}(\Phi u_{k+1/2})), \\ v_{k+1} = (I - \mathrm{diag}(Qu_{k+1/2}))^{-1}(e - (\Psi v_{k+1/2}) \circ u_{k+1/2} + (I - \mathrm{diag}(Pv_{k+1/2}))^{-1}(\Psi v_{k+1/2})). \end{cases} \tag{14}$$

We refer (14) to two iteration steps of TSRN.

It is clear from these computational schemes that NBJ, NBGS, and TSRN have the same computational cost, about $8n^2$ flops, for two iteration steps, while the iteration (12) requires $12n^2$ flops for two steps. Moreover, NBJ, TSRN, and the iteration (12) are more feasible in parallel than NBGS.

4. Numerical experiments

In this section, we present a numerical example to confirm the convergence results. Let ITER denote the iteration scheme (12). We compare ITER with NBGS.

Define

$$\mathrm{ERR}_k = \max\left\{ \frac{\|u_{k+1} - u_k\|_2}{\|u_{k+1}\|_2}, \frac{\|v_{k+1} - v_k\|_2}{\|v_{k+1}\|_2} \right\},$$

where $\|\cdot\|_2$ is the 2-norm for a vector.

All the numerical experiments are performed in Matlab on a PC with the usual double precision, where the floating point relative accuracy is $2.22 \cdot 10^{-16}$.

We consider (1) with $n = 32$. As in [2], the constants c_i and ω_i are given by a numerical quadrature formula on the interval [0, 1], which is obtained by dividing [0, 1] into $n/4$ subinterval of equal length and applying a Gauss–Legendre quadrature with 4 nodes to each subinterval.

We test three values of (α, c) taken to be (0.1, 0.9), (0.01, 0.99), (0, 1).

Figs. 1–3 depict the numerical results for ITER and NBGS. It is easy to see that although NBGS is slightly faster than ITER, they have almost the same convergence rate. Moreover, the asymptotic convergence rate of these two methods is linear when $(\alpha, c) \neq (0, 1)$ and sublinear when $(\alpha, c) = (0, 1)$.

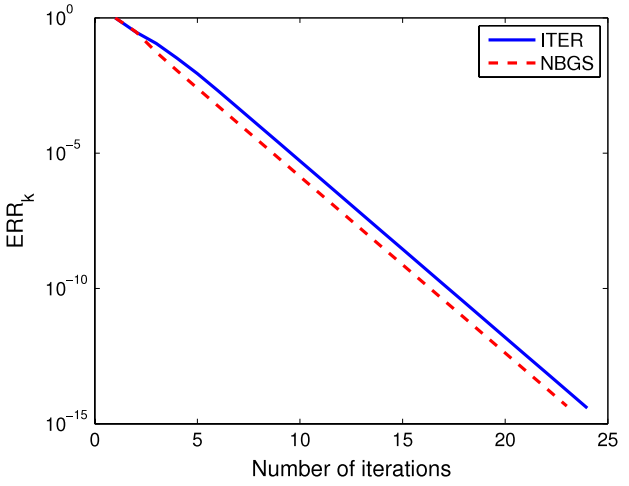

Fig. 1. $(\alpha, c) = (0.1, 0.9)$.

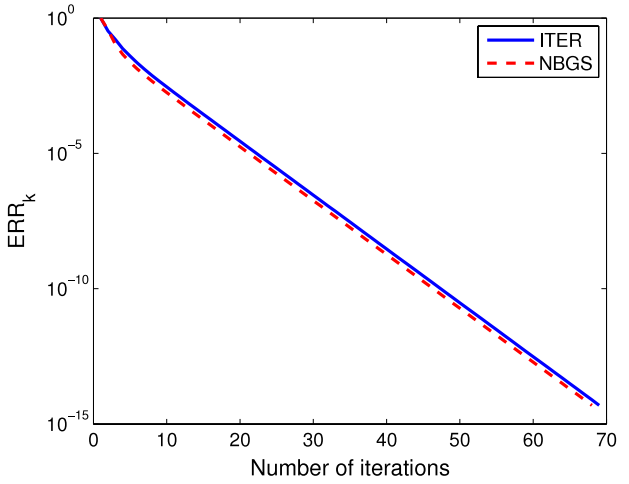

Fig. 2. $(\alpha, c) = (0.01, 0.99)$.

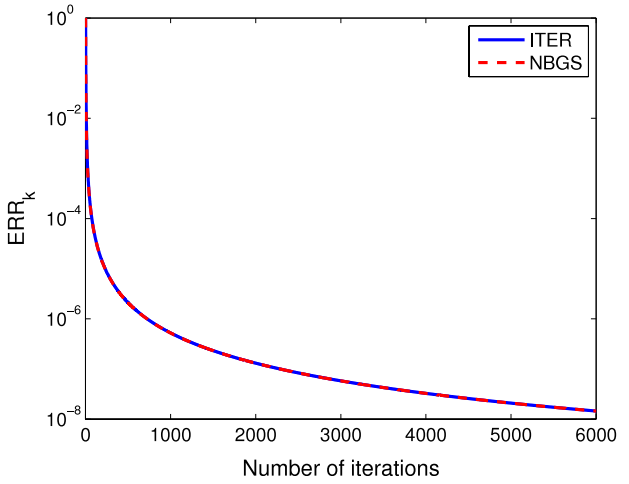

Fig. 3. $(\alpha, c) = (0, 1)$.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful suggestions, which greatly improved the paper. This work is supported by the National Natural Science Foundation of China under grants 10801048 and 10926150, the Natural Science Foundation of Hunan Province under grants 09JJ6014 and 11JJ4009, the Key Program of the Scientific Research Foundation from Education Bureau of Hunan Province under grant 09A033, the Scientific Research Foundation of Education Bureau of Hunan Province for Outstanding Young Scholars in University under grant 10B038, the Science and Technology Planning Project of Hunan Province under grant 2010JT4042, and the Young Core Teacher Foundation of Hunan Province in University.

References

[1] J. Juang, Existence of algebraic matrix Riccati equations arising in transport theory, Linear Algebra Appl. 230 (1995) 89–100.
[2] J. Juang, W.W. Lin, Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices, SIAM J. Matrix Anal. Appl. 20 (1998) 228–243.
[3] L.Z. Lu, Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory, SIAM J. Matrix Anal. Appl. 26 (2005) 679–685.
[4] L. Bao, Y. Lin, Y. Wei, A modified simple iterative method for nonsymmetric algebraic Riccati equations arising in transport theory, Appl. Math. Comput. 181 (2006) 1499–1504.
[5] Z.Z. Bai, Y.H. Gao, L.Z. Lu, Fast iterative schemes for nonsymmetric algebraic Riccati equations arising from transport theory, SIAM J. Sci. Comput. 30 (2008) 804–818.
[6] S. Wu, C. Huang, Two-step relaxation Newton method for nonsymmetric algebraic Riccati equations arising from transport theory, Math. Probl. Eng. 12 (2009) 1–17.
[7] C.H. Guo, W.W. Lin, Convergence rates of some iterative methods for nonsymmetric algebraic Riccati equations arising in transport theory, Linear Algebra Appl. 432 (2010) 283–291.
[8] D.A. Bini, B. Iannazzo, F. Poloni, A fast Newton's method for a nonsymmetric algebraic Riccati equation, SIAM J. Matrix Anal. Appl. 30 (2008) 276–290.
[9] C.H. Guo, B. Iannazzo, B. Meini, On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation, SIAM J. Matrix Anal. Appl. 29 (2007) 1083–1100.
[10] V. Mehrmann, H. Xu, Explicit solutions for a Riccati equation from transport theory, SIAM J. Matrix Anal. Appl. 30 (2008) 1339–1357.
[11] Y. Lin, A class of iterative methods for solving nonsymmetric algebraic Riccati equations arising in transport theory, Comput. Math. Appl. 56 (2008) 3046–3051.