Electronic Transactions on Numerical Analysis.
Volume 40, pp. 489-506, 2013.
Copyright © 2013, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

VECTOR EXTRAPOLATION APPLIED TO ALGEBRAIC RICCATI EQUATIONS ARISING IN TRANSPORT THEORY[*]

ROLA EL-MOALLEM[†] AND HASSANE SADOK[‡]

Abstract. We apply the reduced rank extrapolation method (RRE) to an iterative method for computing the minimal positive solution of a nonsymmetric algebraic Riccati equation that arises in transport theory. The computations yield the minimal positive solution of a vector equation, which is derived from the special form of solutions of the Riccati equation and by exploiting the special structure of the coefficient matrices of the Riccati equation. Numerical experiments and comparisons illustrate the effectiveness of the new approach.

Key words. nonsymmetric algebraic Riccati equation, transport theory, minimal positive solution, iterative methods, vector sequences, polynomial vector extrapolation methods, convergence acceleration, reduced rank extrapolation.

AMS subject classifications. 15A24, 65F10, 65B05

1. Introduction.

An important problem that arises in different areas of science and engineering is the computation of limits of sequences of vectors $x_0, x_1, \ldots$, where x_i are complex N-vectors with N large. Such sequences may result from iterative methods or perturbation techniques and may converge very slowly. This is the case, for example, when they result from finite-difference or finite-element discretizations of continuum problems, where rates of convergence become worse as the mesh size decreases. Vector extrapolation methods can be applied to such vector sequences. These methods transform a sequence of vectors generated by some process to a new sequence which converges faster, without requiring explicit knowledge of the sequence generator.

Our concern is a special kind of nonsymmetric algebraic Riccati equation (NARE) that arises in transport theory. We will apply a polynomial vector extrapolation method [7, 34], namely RRE, to a vector sequence produced by an iterative method for computing the minimal positive solution of NARE.

This paper is organized as follows. In Section 2, we introduce a specific kind of NARE that arises in transport theory. We will review some iterative methods which have been proven to be efficient for solving these kinds of equations. We will focus on the method of Lin [24], and we will make a small modification, similar to that of NBGS in [1] to accelerate it. Section 3 is devoted to polynomial vector extrapolation methods, namely the reduced rank extrapolation (RRE). RRE is an efficient convergence accelerator and a short review of this method will be given. An application of the RRE method to the iterative method of Lin [24] is considered followed by a numerical example in Section 4 which shows the effectiveness of our approach.

[*]Received July 25, 2013. Accepted for publication October 1, 2013. Published online December 28, 2013. Recommended by L. Reichel.

[†]Laboratoire Paul Painlevé, Université Lille 1 - Sciences et Technologies, Batiment M2, 59 655 Villeneuve d'Ascq Cedex, France (rola.el-moallem@math.univ-lille1.fr).

[‡]Laboratoire de Mathématiques Pure et Appliquées, Université du Littoral, Centre Universitaire de la Mi-Voix, Batiment H. Poincarré, 50 Rue F. Buisson, BP 699, 62228 Calais Cedex, France (sadok@lmpa.univ-littoral.fr).

2. NARE.

Nonsymmetric algebraic Riccati equations appear in transport theory, where a variation of the usual one-group neutron transport equation [3, 9, 12, 22] is formulated as

$$\begin{aligned} \left[(\mu+\alpha)\frac{\partial}{\partial x}+1\right]\varphi(x,\mu) &= \frac{c}{2}\int_{-1}^{1}\varphi(x,w)\,dw\,, \\ \varphi(0,\mu) &= f(\mu)\,, \qquad \mu>-\alpha\,,\ |\mu|\le 1\,, \\ \lim_{x\to\infty}\varphi(x,\mu) &= 0\,, \end{aligned} \tag{2.1}$$

where $\varphi(x)$ is the neutron flux, α ($0\le\alpha<1$) is an angular shift, and c is the average of the total number of particles emerging from a collision, which is assumed to be conserved, i.e., $0<c\le 1$.

The scattering function $L:[-\alpha,1]\times[\alpha,1]\to\mathbb{R}$ for particle transport in the half-space can be derived from (2.1) and satisfies the integro-differential equation

$$\left(\frac{1}{\mu+\alpha}+\frac{1}{y-\alpha}\right)L(\mu,y)=c\left(1+\frac{1}{2}\int_{-\alpha}^{1}\frac{L(\omega,y)}{\omega+\alpha}d\,\omega\right)\left(1+\frac{1}{2}\int_{\alpha}^{1}\frac{L(\mu,\omega)}{\omega-\alpha}d\,\omega\right), \tag{2.2}$$

where $(\mu,y)\in[-\alpha,1]\times[\alpha,1]$ and $(\alpha,c)\in(0,1)^2$ are pairs of nonnegative constants; see the appendix in [19].

When $c=0$ or $\alpha=1$, the integro-differential equation (2.2) has a trivial solution [3]. However, when $0\le\alpha<1$ and $0<c\le 1$, it has a unique, positive, uniformly bounded, and globally defined solution [19].

Discretization of the integro-differential equation (2.2) by a numerical quadrature formula on the interval $[0,1]$ yields an algebraic matrix Riccati equation.

Nonsymmetric algebraic Riccati equations (NARE) are quadratic matrix equations of the general form

$$XCX-XD-AX+B=0. \tag{2.3}$$

where the coefficient matrices $A,B,C,$ and $D\in\mathbb{R}^{n\times n}$.

Let $\{\omega_i\}_{i=1}^n$ and $\{c_i\}_{i=1}^n$ denote the sets of nodes and weights, respectively, of the specific quadrature rule that is used on the interval [0, 1]. This quadrature rule is obtained by dividing the interval into $n/4$ subintervals of equal length and applying a Gauss-Legendre quadrature with 4 nodes to each subinterval. These typically satisfy:

$$c_1,\dots,c_n>0,\quad \sum_{i=1}^{n}c_i=1\,,\quad \text{and}\quad 1>\omega_1>\omega_2>\dots>\omega_n>0. \tag{2.4}$$

We are interested in a special case of NARE where

$$A=\Delta-eq^T,\ B=ee^T,\ C=qq^T,\ D=\Gamma-qe^T, \tag{2.5}$$

and

$$\begin{aligned} e&=[1,1,...,1]^T, && \\ q&=[q_1,q_2,...,q_n]^T, && q_i=\frac{c_i}{2\omega_i},\ q_i>0, \\ \Delta&=\mathrm{diag}([\delta_1,\delta_2,...,\delta_n]), && \delta_i=\frac{1}{c\omega_i(1+\alpha)}, \\ \Gamma&=\mathrm{diag}([\gamma_1,\gamma_2,...,\gamma_n]), && \gamma_i=\frac{1}{c\omega_i(1-\alpha)}, \end{aligned} \tag{2.6}$$

where $\mathrm{diag}(\cdot)$ applied to a vector denotes a square diagonal matrix with the elements of the vector on its diagonal. Here, $0 \leq \alpha < 1$ and $0 < c \leq 1$. As a consequence of (2.4), we have

$$0 < \delta_1 < \delta_2 < \ldots \delta_n \text{ and } 0 < \gamma_1 < \gamma_2 < \ldots < \gamma_n.$$

In addition,

$$\begin{array}{ll} \gamma_i \leq \delta_i & \text{for } \alpha \leq 0, \\ \gamma_i > \delta_i & \text{for } \alpha \neq 0, \end{array}$$

for $i = 1, 2, \ldots, n$.

The term nonsymmetric distinguishes this case from the widely studied continuous-time algebraic Riccati equations (CARE), defined by the quadratic matrix equation

$$XCX - AX - XA^T + B = 0,$$

where B and C are symmetric. For a comprehensive analysis of CARE; see [5, 23, 27].

Nonsymmetric algebraic Riccati equations arise in many fields such as transport theory when dealing with particle transfer (or radiative transfer), Wiener-Hopf factorization of Markov chains, nuclear physics, applied probability, engineering, control theory, etc. In this paper, we are interested in special Riccati equations that arise in transport theory with A, B, C, and D given as in (2.5).

The existence of nonnegative solutions of (2.3) was shown by Juang [18] using degree theory. It is shown [18, 21] that the Riccati equation (2.3) has two entry-wise positive solutions $X = [x_{ij}]$ and $Y = [y_{ij}]$ in $\mathbb{R}^{n\times n}$ which satisfy $X \leq Y$, where we use the notation $X \leq Y$ if $x_{ij} \leq y_{ij}$ for all $i, j = 1, ..., n$. In applications from transport theory, only the smaller one of the two positive solutions is of interest and is physically meaningful. Some iterative procedures [20] were developed to find both nonnegative solutions.

2.1. Obtaining the minimal positive solution of NARE by a vector equation. It has been shown by Lu [25] that the minimal positive solution of a NARE can be obtained by computing the minimal positive solution of a vector equation. This vector equation is derived from a special form of solutions of the Riccati equation and by exploiting the special structure of the coefficient matrices of the Riccati equation.

Rewrite the NARE (2.3)–(2.5) as

$$\Delta X + XD = (Xq + e)(q^T X + e^T)$$

and let

$$u = Xq + e \qquad \text{and} \qquad v^T = q^T X + e^T. \tag{2.7}$$

It has been shown in [18, 21] that any solution of (2.3) must be of the form

$$X = T \circ (uv^T) = (uv^T) \circ T, \tag{2.8}$$

where $\circ$ denotes the Hadamard product defined by $A \circ B = [a_{ij}b_{ij}]$ for any two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, T is the Cauchy matrix, $T = [t_{i,j}] = [1/(\delta_i + \gamma_j)]$, $X = [x_{i,j}]$, $u = [u_1, u_2, \ldots, u_n]^T$ and $v = [v_1, v_2, \ldots, v_n]^T$.

Finding the minimal positive solution of (2.3) requires finding two positive vectors u and v in (2.8). Substituting (2.8) into (2.7) gives the vector equation

$$\begin{aligned} u &= u \circ (Pv) + e, \\ v &= v \circ (Qu) + e, \end{aligned} \tag{2.9}$$

where

$$P = [P_{ij}] = [\frac{q_j}{\delta_i + \gamma_j}] = T\,\mathrm{diag}(q) \tag{2.10}$$

and

$$Q = [Q_{ij}] = [\frac{q_j}{\delta_j + \gamma_i}] = T^T\,\mathrm{diag}(q). \tag{2.11}$$

The minimal positive solution of (2.3) can be obtained by computing the minimal positive solution of the vector equation (2.9). This vector equation can be expressed as

$$\begin{array}{l} f(u,v) := u - u \circ (Pv) - e = 0, \\ g(u,v) := v - v \circ (Qu) - e = 0 \end{array} .$$

which is a system of nonlinear equations having the vector pair (u, v) as a positive solution. Based on the above, Lu [25] defined an iterative method for the solution of (2.9).

2.1.1. The iterative method of Lu.

Lu defined an iterative scheme to find the positive vectors u and v,

$$\begin{array}{l} u^{(k+1)} = u^{(k)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} = v^{(k)} \circ (Qu^{(k)}) + e, \quad \text{for } k = 0, 1, \ldots, \\ (u^{(0)}, v^{(0)}) = (0, 0), \end{array} \tag{2.12}$$

satisfying a certain stopping criterion. For all $0 \leq \alpha < 1$ and $0 < c \leq 1$, the sequence $\{(u^{(k)}, v^{(k)})\}$ defined by (2.12) is strictly monotonically increasing, bounded above, and thus converging; see [25] for the proof. Each iteration costs approximately $4n^2$ flops.

The minimal positive solution of (2.3) can be computed by

$$X^* = T \circ (u^*(v^*)^T),$$

where (u^*, v^*) is the limit of $(u^{(k)}, v^{(k)})$ defined by the iteration scheme (2.12).

2.1.2. A modified iterative method.

Bao et al. [2] proposed a modified version of the iterative method of Lu [25]. They noticed that since $u^{(k+1)}$ is obtained before $v^{(k+1)}$, it should be a better approximation to u^* than $u^{(k)}$. Consequently, $u^{(k)}$ is replaced with $u^{(k+1)}$ in the equation for $v^{(k+1)}$ of the iteration scheme (2.12).

The modified iteration is as follows

$$\begin{array}{l} \tilde{u}^{(k+1)} = \tilde{u}^{(k)} \circ (P\tilde{v}^{(k)}) + e, \\ \tilde{v}^{(k+1)} = \tilde{v}^{(k)} \circ (Q\tilde{u}^{(k+1)}) + e, \quad \text{for } k = 0, 1, \ldots, \\ \tilde{u}^{(0)} = \tilde{v}^{(0)} = 0. \end{array} \tag{2.13}$$

The monotonic convergence of the modified iteration scheme (2.13) is illustrated in [2]. Also, it is shown how the minimal positive solution X^* of the NARE (2.3) can be computed from

$$X^* = T \circ (u^*(v^*)^T),$$

where (u^*, v^*) is the limit of $(\tilde{u}^{(k)}, \tilde{v}^{(k)})$ defined by the modified iteration (2.13).

Since it was also shown that $u^{(k)} < \tilde{u}^{(k)}$ and $v^{(k)} < \tilde{v}^{(k)}$ for $k \geq 3$, $(\tilde{u}^{(k)}, \tilde{v}^{(k)})$ has the same limit (u^*, v^*) as $(u^{(k)}, v^{(k)})$. Also, this proves how the modified iteration scheme

ETNA
Kent State University
http://etna.math.kent.edu

(2.13) is more efficient than (2.12). The cost of every iteration step is approximately $4n^2$ flops.

Two accelerated variants of the iterative scheme (2.12) of Lu, proposed in [1], are the nonlinear block Jacobi (NBJ) iteration scheme

$$\begin{aligned} u^{(k+1)} &= u^{(k+1)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} &= v^{(k+1)} \circ (Qu^{(k)}) + e, \quad \text{for } k = 0, 1, \dots, \end{aligned} \tag{2.14}$$

and the nonlinear block Gauss-Seidel (NBGS) iteration scheme

$$\begin{aligned} u^{(k+1)} &= u^{(k+1)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} &= v^{(k+1)} \circ (Qu^{(k+1)}) + e, \quad \text{for } k = 0, 1, \dots. \end{aligned} \tag{2.15}$$

For both NBJ and NBGS, the sequence $\{(u^{(k)}, v^{(k)})\}$ is shown in [1] to be strictly monotonically increasing and convergent to the minimal positive solution (u^*, v^*) of the vector equations (2.9). NBJ and NBGS have the same computational costs at every iteration step (approximately $4n^2$ flops). Both are effective solvers of NAREs arising in transport theory, but NBGS is more efficient than NBJ in applications. In particular, in terms of the asymptotic rates of convergence, the NBGS method is twice as fast as the NBJ method; see [15, Theorem 5] which explains the numerical results presented in [1], where the number of iterations required for NBGS is half of that for NBJ.

In conclusion, the four fixed-point iterations (2.12), (2.13), (2.14), and (2.15) are easy to use and share the same low complexity at every iteration. However, NBGS was proved in [15] to be the fastest among these methods in terms of the asymptotic rate of convergence when $(\alpha, c) \neq (0, 1)$, although being sublinear when $(\alpha, c) = (0, 1)$. The sublinear convergence, which takes place when the Jacobian at the required solution is singular, is transformed into a quadratic convergence by means of a Newton method proposed in [4].

2.1.3. The Newton method. Guo and Laub proposed in [14] an application of the Newton method to the Riccati equation (2.3). This method has an order of complexity $O(n^3)$ per step and yields quadratic convergence. Lu [26] instead proposed an algorithm that consists of iteration (2.12) combined with a Newton iteration. This scheme is simple and more efficient than applying the Newton method directly to (2.3). Another approach was proposed by Bini et al. [4] who apply a fast Newton method to the NARE (2.3). It consists of an algorithm which performs the Newton iteration in $O(n^2)$ flops. This approach relies on a modification of the fast LU factorization algorithm for Cauchy-like matrices proposed by Gohberg et al. [13]. The same idea reduces the cost of Newton's algorithm proposed by Lu [26] from $O(n^3)$ to $O(n^2)$ while preserving the quadratic convergence in the generic case.

2.1.4. The iterative method of Lin. This method was proposed in [24]. Let $w = [u^T, v^T]^T$ and reformulate the vector equation (2.9) as

$$f(w) := w - w \circ Mw - e = 0, \tag{2.16}$$

where,

$$M = \begin{bmatrix} 0 & P \\ Q & 0 \end{bmatrix}.$$

The Jacobian matrix $J(f, w)$ of $f(w)$ for any $w \in \mathbb{R}^{2n}$ is given by

$$J(f, w) = I_{2n} - N(w),$$

where

$$N(w) = \begin{bmatrix} \operatorname{diag}(Pv) & \operatorname{diag}(u)P \\ \operatorname{diag}(v)Q & \operatorname{diag}(Qu) \end{bmatrix}.$$

Lin constructed a class of iterative methods to solve the vector equation (2.16) based on the following iterative scheme

$$w^{(k+1)} := w^{(k)} - T_k^{-1} f(w^{(k)}), \quad \text{for } k = 0, 1, 2, \dots, \tag{2.17}$$

where T_k is chosen to approximate $J(f, w^{(k)})$ and $T_k \geq J(f, w^{(k)})$. Note that when $T_k = J(f, w^{(k)})$, the Newton method results. A disadvantage of the Newton method is that it requires an LU factorization of the Jacobian matrix $J(f, w^{(k)})$ at each iteration step to obtain the new approximation to w^*. This costs $O(n^3)$ operations per step.

For any $w^{(k)} = [(u^{(k)})^T, (v^{(k)})^T]^T \in \mathbb{R}^{2n}$, Lin [24] proposed the choice

$$T_k = I_{2n} - \begin{bmatrix} \operatorname{diag}(Pv^{(k)}) & 0 \\ 0 & \operatorname{diag}(Qu^{(k)}) \end{bmatrix}. \tag{2.18}$$

Substituting (2.18) into (2.17) gives

$$w^{(k+1)} = \begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} := \begin{bmatrix} (I_n - \operatorname{diag}(Pv^{(k)}))^{-1} e \\ (I_n - \operatorname{diag}(Qu^{(k)}))^{-1} e \end{bmatrix}. \tag{2.19}$$

Convergence of the iterative method (2.19) was examined in [24]. It was shown that the convergence is sublinear as (α, c) tends to $(0, 1)$ and linear when $(\alpha, c) = (0, 1)$. Since $(I_n - \operatorname{diag}(Pv^{(k)}))$ and $(I_n - \operatorname{diag}(Qu^{(k)}))$ are diagonal matrices, the computational cost of each iteration step is approximately $4n^2$ flops. A numerical comparison shows how the iterative method of Lin converges faster than the modified iterative method of Lu (2.13) and the Newton method.

2.1.5. A modification of the iterative method of Lin. A modification of the iterative scheme (2.19) that is an analog of the NBGS method [1] is proposed. In the iterative scheme (2.19), $u^{(k+1)}$ is computed before $v^{(k+1)}$. Therefore, $u^{(k+1)}$ should be a better approximation of u^* than $u^{(k)}$. Replacing $u^{(k)}$ by $u^{(k+1)}$ in the equation for $v^{(k+1)}$ in (2.19) leads to the modified iteration scheme

$$w^{(k+1)} = \begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} := \begin{bmatrix} (I_n - \operatorname{diag}(Pv^{(k)}))^{-1} e \\ (I_n - \operatorname{diag}(Qu^{(k+1)}))^{-1} e \end{bmatrix}. \tag{2.20}$$

For a matrix $G \in \mathbb{R}^{n \times n}$, we define the following function

$$\begin{aligned} \Phi_G \; : \mathbb{R}^n &\longrightarrow \mathbb{R}^n, \\ t &\longrightarrow (I_n - \operatorname{diag}(Gt))^{-1} e. \end{aligned}$$

Then iteration (2.20) can be expressed as

$$w^{(k+1)} = \begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} := \begin{bmatrix} \Phi_P(v^{(k)}) \\ \Phi_Q(u^{(k+1)}) \end{bmatrix}. \tag{2.21}$$

LEMMA 2.1. *Let (u^*, v^*) be the minimal positive solution of* (2.20). *Then, $u^* > e$ and $v^* > e$.*

Proof. Let (u^*, v^*) be the minimal positive solution of (2.20). Then

$$u^* = (I_n - \mathrm{diag}(Pv^*))^{-1}e > e > 0,$$
$$v^* = (I_n - \mathrm{diag}(Qu^*))^{-1}e > e > 0,$$

since P and Q are positive matrices. □

THEOREM 2.2. *Given the sequence of vectors generated by* (2.20) *with initial vector* $(u^{(0)}, v^{(0)}) = (0, 0)$*, let* (u^*, v^*) *be the minimal positive solution of* (2.20)*. Then the sequence* $\{(u^{(k)}, v^{(k)})\}$ *is strictly monotonically increasing, bounded above and thus convergent.*

Proof. To show this, we have to prove component-wise that

(i) $0 \leq u^{(k)} < u^{(k+1)} < u^*$ and $0 \leq v^{(k)} < v^{(k+1)} < v^*,\ k \geq 1;$
(ii) $\lim\limits_{k\to\infty} u^{(k)} = u^*$ and $\lim\limits_{k\to\infty} v^{(k)} = v^*$.

We start by proving (i) by induction. Let $(u^{(0)}, v^{(0)}) = (0, 0)$ be a starting point for this method. For $k = 0$, using (2.20) and Lemma 2.1, we get

$$u^{(0)} = 0 < e = u^{(1)} < u^*.$$

Also,

$$v^{(1)} = [I_n - \mathrm{diag}(Qu^{(1)})]^{-1}e = [I_n - \mathrm{diag}(Qe)]^{-1}e < [I_n - \mathrm{diag}(Qv^*)]^{-1}e$$

and, therefore,

$$v^{(0)} = 0 < e = v^{(1)} < v^*.$$

This shows that (i) holds for $k = 0$. Now, suppose that (i) holds for a positive integer k, i.e., we have

$$0 \leq u^{(k)} < u^{(k+1)} < u^* \qquad \text{and} \qquad 0 \leq v^{(k)} < v^{(k+1)} < v^*, \text{ for } k \geq 1.$$

Using (2.20) and the fact that $v^{(k)} < v^{(k+1)}$, we have

$$\begin{aligned}[I_n - \mathrm{diag}(Pv^{(k+1)})]u^{(k+2)} = e &= [I_n - \mathrm{diag}(Pv^{(k)})]u^{(k+1)} \\ &> [I_n - \mathrm{diag}(Pv^{(k+1)})]u^{(k+1)}.\end{aligned}$$

Since $I_n - \mathrm{diag}(Pv^{(k+1)}) > I_n - \mathrm{diag}(Pv^*) > 0$, using Lemma 2.1 and the fact that $v^{(k+1)} < v^*$, we get

$$0 \leq u^{(k+1)} < u^{(k+2)}.$$

Also,

$$\begin{aligned}[I_n - \mathrm{diag}(Pv^{(k+1)})]u^{(k+2)} = e &= [I_n - \mathrm{diag}(Pv^*)]u^* \\ &< [I_n - \mathrm{diag}(Pv^{(k+1)})]u^*.\end{aligned}$$

Therefore, $u^{(k+2)} < u^*$ holds, and consequently (i) holds for $(k + 1)$.
In conclusion, (i) is shown by induction.

The proof of (ii) can be done via (i) which provides the existence of two positive vectors $(\hat{u}^*, \hat{v}^*)$, with $0 < \hat{u}^* < u^*$ and $0 < \hat{v}^* < v^*$, satisfying

$$\lim_{k\to\infty} u^{(k)} = \hat{u}^* \qquad \text{and} \qquad \lim_{k\to\infty} v^{(k)} = \hat{v}^*,$$

and

$$\hat{u}^* = \Phi_P(\hat{v}^*),$$
$$\hat{v}^* = \Phi_Q(\hat{u}^*),$$

i.e., $(\hat{u}^*, \hat{v}^*)$ is a positive solution of (2.20). Due to the minimal property of (u^*, v^*) and the comparative property of $(\hat{u}^*, \hat{v}^*)$ with (u^*, v^*), it must hold that $\hat{u}^* = u^*$ and $\hat{v}^* = v^*$. □

We consider the case when $\alpha = 0$ and $c = 1$, which is referred to as the critical case. Looking back at equations (2.6), it can be seen that when $\alpha = 0$, we have $\delta_i = \gamma_i = \frac{1}{\omega_i}$ and consequently $\Delta = \Gamma$. We have $T = [t_{i,j}] = [\frac{1}{\delta_i+\gamma_j}]$ and since $\delta_i = \gamma_i$, it follows that $T = [\frac{1}{\delta_i+\delta_j}]$ and T is symmetric ($T = T^T$). In view of (2.10) and (2.11), we get $P = Q$ and consequently $\Phi_P = \Phi_Q$.

Let K and L be two matrices in $\mathbb{R}^{n\times n}$ and define the following transformation

$$\begin{aligned} \Phi_{K,L} \ : \mathbb{R}^n &\longrightarrow \ \mathbb{R}^n, \\ t \ &\longrightarrow \ \Phi_K(\Phi_L(t)). \end{aligned}$$

Then iteration (2.21) becomes

$$\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} = \begin{bmatrix} \Phi_{P,Q}(u^{(k)}) \\ \Phi_{Q,P}(v^{(k)}) \end{bmatrix}.$$

And for $\Phi_P = \Phi_Q$,

$$\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} = \begin{bmatrix} \Phi_{P,P}(u^{(k)}) \\ \Phi_{P,P}(v^{(k)}) \end{bmatrix}.$$

Starting with equal initial vectors $u^{(0)} = v^{(0)} = 0$, it can be easily seen by induction that the sequences $\{u^{(k)}\}_{k\in\mathbb{N}}$ and $\{v^{(k)}\}_{k\in\mathbb{N}}$ are equal. This implies that it is enough to compute one of the vector sequences, for example $\{u^{(k)}\}_{k\in\mathbb{N}}$, and the solution of NARE then can be calculated by

$$X \ = \ T \circ (u^*(u^*)^T),$$

where u^* denotes the limit of $u^{(k)}$. Thus, only half of the computational work of (2.21) is needed.

Now, we consider the calculation of the Jacobian matrix of the modified iteration (2.21). Let

$$\begin{aligned} \Phi \ : \mathbb{R}^n \times \mathbb{R}^n &\longrightarrow \ \mathbb{R}^n \times \mathbb{R}^n \ , \\ (u, v) \ &\longrightarrow \ (\Phi_{P,Q}(u), \Phi_{Q,P}(v)) \end{aligned}$$

be the transformation describing the iteration (2.21). Then the iterative scheme (2.21) can be written as

$$\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} \ := \ \begin{bmatrix} \Phi_{P,Q}(u^{(k)}) \\ \Phi_{Q,P}(v^{(k)}) \end{bmatrix}.$$

The Jacobian matrix is given by

$$\begin{aligned} J(\Phi, (u, v)) &= \begin{bmatrix} J(\Phi_{P,Q}, (u, v)) \\ J(\Phi_{Q,P}, (u, v)) \end{bmatrix} \\ &= \begin{bmatrix} J(\Phi_P, \Phi_Q(u))J(\Phi_Q, u) & 0 \\ 0 & J(\Phi_Q, \Phi_P(v))J(\Phi_P, v) \end{bmatrix}. \end{aligned}$$

At the solution (u^*, v^*), the Jacobian is of the form

$$J(\Phi,(u^*,v^*)) = \begin{bmatrix} J(\Phi_P,v^*)J(\Phi_Q,u^*) & 0 \\ 0 & J(\Phi_Q,u^*)J(\Phi_P,v^*) \end{bmatrix}, \tag{2.22}$$

because $\Phi_Q(u^*) = v^*$ and $\Phi_P(v^*) = u^*$. We have

$$(I_n - \mathrm{diag}(Pv^*))^{-1}e = \begin{bmatrix} \frac{1}{1-\sum_{j=1}^n P_{1j}v_j^*} \\ \frac{1}{1-\sum_{j=1}^n P_{2j}v_j^*} \\ \vdots \\ \frac{1}{1-\sum_{j=1}^n P_{nj}v_j^*} \end{bmatrix}.$$

Therefore,

$$J(\Phi_P,v^*) = \begin{bmatrix} \frac{P_{11}}{(1-\sum_{j=1}^n P_{1j}v_j^*)^2} & \cdots & \frac{P_{1n}}{(1-\sum_{j=1}^n P_{1j}v_j^*)^2} \\ \vdots & \ddots & \vdots \\ \frac{P_{n1}}{(1-\sum_{j=1}^n P_{nj}v_j^*)^2} & \cdots & \frac{P_{nn}}{(1-\sum_{j=1}^n P_{nj}v_j^*)^2} \end{bmatrix} = \{(I_n - \mathrm{diag}(Pv^*))^{-1}\}^2 P.$$

Define

$$\begin{aligned} K \ : \mathbb{R}^{n\times n} \times \mathbb{R}^n &\longrightarrow \mathbb{R}^{n\times n}, \\ (Y,t) \quad &\longrightarrow \{(I_n - \mathrm{diag}(Yt))^{-1}\}^2 Y. \end{aligned}$$

Then

$$J(\Phi_P,v^*) = K(P,v^*). \tag{2.23}$$

Similarly, we have

$$J(\Phi_Q,u^*) = K(Q,u^*). \tag{2.24}$$

Substituting (2.23) and (2.24) into (2.22), the Jacobian matrix at the solution becomes

$$J(\Phi,(u^*,v^*)) = \begin{bmatrix} K(P,v^*)K(Q,u^*) & 0 \\ 0 & K(Q,u^*)K(P,v^*) \end{bmatrix}. \tag{2.25}$$

3. Vector extrapolation methods [7, 34] . The most popular vector extrapolation methods are the minimal polynomial extrapolation (MPE) method by Cabay and Jackson [8], the reduced rank extrapolation (RRE) method by Eddy [10] and Mesina [28], the modified minimal polynomial extrapolation (MMPE) method by Sidi et al. [33], Brezinski [6], and Pugachev [29], the topological ϵ-algorithm (TEA) by Brezinski [6], and the vector ϵ-algorithm (VEA) by Wynn [35, 36]. These methods do not require explicit knowledge of how the sequence is generated and can be applied to the solution of linear and nonlinear systems of equations. Several recursive algorithms for implementing these methods are presented in [6, 10, 11, 16]. A numerical comparison of vector extrapolation methods and their applications can be found in [17, 34].

We are interested in applying the RRE method to the iteration (2.20) to accelerate its convergence. See Table 4.1 for a comparison between RRE, MPE, and MMPE which explains the choice of using RRE. Throughout this paper, we denote by $(\cdot,\cdot)$ the Euclidean inner product in $\mathbb{R}^N$ and by $\|\cdot\|$ the corresponding norm.

ETNA
Kent State University
http://etna.math.kent.edu

3.1. Review of RRE [34]. Let $\{s^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of vectors in $\mathbb{R}^N$. Define the first and the second forward differences of $s^{(k)}$ by

$$\Delta s^{(k)} = s^{(k+1)} - s^{(k)} \quad \text{and} \quad \Delta^2 s^{(k)} = \Delta s^{(k+1)} - \Delta s^{(k)}\ , \ k = 0, 1, \ldots ,$$

When applied to the vector sequence $\{s^{(k)}\}_{k\in\mathbb{N}}$, the RRE method produces an approximation $t^{(k)}$ of the limit or the antilimit of $\{s^{(k)}\}_{k\in\mathbb{N}}$; see [30]. The approximation $t^{(k)}$ is defined by

$$t^{(k)} = \sum_{j=0}^{k} \eta_j^{(k)} s^{(j)}, \tag{3.1}$$

where

$$\sum_{j=0}^{k} \eta_j^{(k)} = 1 \tag{3.2}$$

and

$$\sum_{j=0}^{k} \beta_{i,j}\eta_j^{(k)} = 0, \qquad \text{for} \quad i = 0, 1, ..., k-1. \tag{3.3}$$

The scalars $\beta_{i,j} \in \mathbb{R}$ are defined by

$$\beta_{i,j} = (\Delta^2 s^{(i)}, \Delta s^{(j)}),$$

for $i = 0, 1, ..., k-1$ and $j = 0, 1, ..., k$.

It follows from (3.1), (3.2), and (3.3) that $t^{(k)}$ can be expressed as the ratio of two determinants

$$t^{(k)} = \frac{\begin{vmatrix} s^{(0)} & s^{(1)} & \ldots & s^{(k)} \\ \beta_{0,0} & \beta_{0,1} & \ldots & \beta_{0,k} \\ \vdots & \vdots & & \vdots \\ \beta_{k-1,0} & \beta_{k-1,1} & \ldots & \beta_{k-1,k} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \ldots & 1 \\ \beta_{0,0} & \beta_{0,1} & \ldots & \beta_{0,k} \\ \vdots & \vdots & & \vdots \\ \beta_{k-1,0} & \beta_{k-1,1} & \ldots & \beta_{k-1,k} \end{vmatrix}}.$$

Using the following notation

$$\Delta^i S^{(k)} = [\Delta^i s^{(0)}, \ldots, \Delta^i s^{(k-1)}]$$

and Schur complements, $t^{(k)}$ can be written as

$$t^{(k)} = s^{(0)} - \Delta S^{(k)}(\Delta^2 S^{(k)})^{+}\Delta s^{(0)},$$

where $(\Delta^2 S^{(k)})^{+}$ denotes the Moore-Penrose generalized inverse of $\Delta^2 S^{(k)}$ defined by

$$(\Delta^2 S^{(k)})^{+} = ((\Delta^2 S^{(k)})^T \Delta^2 S^{(k)})^{-1}(\Delta^2 S^{(k)})^T.$$

It is clear that $t^{(k)}$ exists and is unique if and only if $\det((\Delta^2 S^{(k)})^T \Delta^2 S^{(k)}) \neq 0$; see [32]. We assume this to be the case. The computation of the approximation $t^{(k)}$ can be carried out by one of the recursive algorithms in [11].

To give an estimate for the residual norm for nonlinear problems, introduce the new transformation

$$\tilde{t}^{(k)} = \sum_{j=0}^{k} \eta_j^{(k)} s^{(j+1)}.$$

We use the generalized residual of $t^{(k)}$ defined in [17]. It is given by

$$\tilde{r}(t^{(k)}) = \tilde{t}^{(k)} - t^{(k)} = \sum_{j=0}^{k} \eta_j \Delta s^{(j)}$$

and can be expressed as

$$\tilde{r}(t^{(k)}) = \Delta s^{(0)} - \Delta^2 S^{(k)} (\Delta^2 S^{(k)})^+ \Delta s^{(0)}.$$

Note that the generalized residual $\tilde{r}(t^{(k)})$ is obtained by projecting $\Delta s^{(k)}$ orthogonally onto the subspace spanned by $\Delta^2 s^{(0)}, \dots, \Delta^2 s^{(k-1)}$. It is an approximate residual unless the sequence is generated linearly, in which case it is the true residual. Therefore, a stopping criterion can be based on $\|\tilde{r}(t^{(k)})\|$.

3.2. Implementation of the RRE method. We follow the description given by Sidi in [31]. An important feature of this method is that it proceeds through the solution of least-squares problems by QR factorization.

Introduce the following notation

$$\Delta S^{(k+1)} = [\Delta s^{(0)}, \dots, \Delta s^{(k)}] \quad \text{and} \quad \eta^{(k)} = (\eta_0^{(k)}, \dots, \eta_k^{(k)})^T.$$

In view of (3.1), (3.2), and (3.3), $\eta_j^{(k)}$ can be determined by solving the overdetermined linear system

$$\Delta S^{(k+1)} \eta^{(k)} = 0$$

by the least-squares method subject to the constraint $\sum_{j=0}^{k} \eta_j^{(k)} = 1$. This leads to minimizing the positive definite quadratic form

$$\eta^{(k)^T} (\Delta S^{(k+1)})^T \Delta S^{(k+1)} \eta^{(k)}$$

subject to the same constraint.

Set $d^{(k)} = [d_0^{(k)}, \dots, d_k^{(k)}]^T$, $\lambda = (\sum_{i=0}^{k} d_i^{(k)})^{-1}$, and $\eta^{(k)} = \lambda d^{(k)}$ ($\eta_i^{(k)} = \lambda d_i^{(k)}$). Then $\eta^{(k)}$ can be computed by solving the linear system of equations

$$(\Delta S^{(k+1)})^T \Delta S^{(k+1)} d^{(k)} = e. \tag{3.4}$$

Assume that $\Delta S^{(k+1)}$ has full rank. Then it has a QR factorization $\Delta S^{(k+1)} = Q^{(k)} R^{(k)}$. This leads to another form of the linear system (3.4),

$$(R^{(k)})^T R^{(k)} d^{(k)} = e.$$

Finally, the approximation $t^{(k)}$ can be expressed as

$$t^{(k)} = s^{(0)} + Q^{(k-1)}(R^{(k-1)}\xi^{(k)}),$$

where

$$\xi^{(k)} = [\xi_0^{(k)}, \xi_1^{(k)}, \ldots, \xi_{k-1}^{(k)}]^T, \quad \xi_0^{(k)} = 1 - \eta_0^{(k)}, \quad \text{and} \quad \xi_j^{(k)} = \xi_{j-1}^{(k)} - \eta_j^{(k)}$$

for $j = 1, \ldots, k-1$.

Another expression of $t^{(k)}$ is given by

$$t^{(k)} = s^{(0)} + \sum_{j=0}^{k-1} \xi_j^{(k)} \Delta s^{(j)} = s^{(0)} + \Delta S^{(k)} \xi^{(k)}. \tag{3.5}$$

Then, using (3.4) and (3.5), the generalized residual $\tilde{r}(t^{(k)})$ can be written as

$$\tilde{r}(t^{(k)}) = \sum_{i=0}^{k} \eta_i^{(k)} \Delta s^{(i)} = \Delta S^{(k+1)} \eta^{(k)}.$$

Note that the QR factorization of $\Delta S^{(k+1)}$ is formed by appending one additional column to $Q^{(k-1)}$ to obtain $Q^{(k)}$, and a corresponding column to $R^{(k-1)}$ to obtain $R^{(k)}$. This QR factorization can be computed inexpensively by applying the modified Gram-Schmidt process (MGS) to the vectors $s^{(0)}, s^{(1)}, \ldots, s^{(k+1)}$; see [31].

The computations for the RRE method are described by Algorithm 1. This algorithm becomes increasingly expensive as k increases, because the work requirement grows quadratically with the number of iteration steps k. The storage requirement grows linearly with k. To avoid this, the RRE algorithm should be restarted periodically every r steps for some integer $r > 1$.

Algorithm 1 Basic RRE algorithm.

```
Step 0  Input:Vectors $s^{(0)}, s^{(1)}, \ldots, s^{(k+1)}$.
Step 1  Compute $\Delta s^{(i)} = s^{(i+1)} - s^{(i)}$,   for $i = 0, 1, \ldots, k$.
        Set $\Delta S^{(k+1)} = [\Delta s^{(0)}, \Delta s^{(1)}, \ldots, \Delta s^{(k)}]$.
        Compute the QR factorization of $\Delta S^{(k+1)}$, namely, $\Delta S^{(k+1)} = Q^{(k)} R^{(k)}$.
Step 2  Solve the linear system
        $(R^{(k)})^T R^{(k)} d^{(k)} = e$; where $d^{(k)} = [d_0^{(k)}, d_1^{(k)}, \ldots, d_k^{(k)}]^T$ and $e = [1, 1, \ldots, 1]^T$.
        (This amounts to solving two upper and lower triangular systems.)
        Set $\lambda = (\sum_{i=0}^{k} d_i^{(k)})^{-1}$, $\lambda \in \mathbb{R}^+$.
        Set $\eta_i^{(k)} = \lambda d_i^{(k)}$,   for $i = 0, 1, \ldots, k$.
Step 3  Compute $\xi^{(k)} = [\xi_0^{(k)}, \xi_1^{(k)}, \ldots, \xi_{k-1}^{(k)}]^T$ where $\xi_0^{(k)} = 1 - \eta_0^{(k)}$
        and $\xi_j^{(k)} = \xi_{j-1}^{(k)} - \eta_j^{(k)}$, $1 \le j \le k-1$.
        Compute $t^{(k)}$ by :$t^{(k)} = s^{(0)} + Q^{(k-1)}(R^{(k-1)}\xi^{(k)})$.
```

3.3. Application of the restarted RRE method to the solution of NARE. We apply a restarted RRE algorithm to the vectors computed by the iteration scheme (2.20) to solve the NARE (2.3)–(2.5). Two approaches for restarted RRE are detailed in Algorithms 2 and 3. Algorithm 2 applies a restarted RRE method directly to the vector sequence $\{w^{(k)}\}_{k\in\mathbb{N}}$, where

$w^{(k)} = [(u^{(k)})^T, (v^{(k)})^T]^T$, while Algorithm 3 applies a restarted RRE method to the vector sequences $\{u^{(k)}\}_{k\in\mathbb{N}}$ and $\{v^{(k)}\}_{k\in\mathbb{N}}$ separately. Both approaches accelerate the convergence of the vector sequences but Figure 3.1 shows that the technique of Algorithm 2 works better, especially near the critical case. For later numerical experiments, we therefore use Algorithm 2.

Algorithm 2 The restarted RRE(r) applied to $\{w^{(k)}\}_k$, r is fixed.

```
Input:$k = 0$, $u^{(0)} = v^{(0)} = 0$, choose an integer $r$.
Then, $w^{(0)} = [(u^{(0)})^T, (v^{(0)})^T]^T = 0$.
For $k = 1, 2, \ldots$,
    $y^{(0)} = u^{(k-1)}$; $z^{(0)} = v^{(k-1)}$; $s^{(0)} = (y^{(0)}, z^{(0)})^T$;
    $y^{(j+1)} = \Phi_P(z^{(j)}), z^{(j+1)} = \Phi_Q(y^{(j+1)}), s^{(j+1)} = (y^{(j+1)}, z^{(j+1)})^T, j = 0, \ldots, r-1$.
    Compute the approximations $t^{(r-1)}$ by applying the RRE Algorithm 1 on the vectors
    $(s^{(0)}, s^{(1)}, \ldots, s^{(r)})$;
    If $t^{(r-1)}$ satisfies accuracy test, stop.
    Else
        $w^{(k)} = (u^{(k)}, v^{(k)})^T = t^{(r-1)}$;
End
```

Algorithm 3 The restarted RRE(r) applied to $\{u^{(k)}\}_k$ and $\{v^{(k)}\}_k$, r is fixed.

```
Input:$k = 0$, $u^{(0)} = v^{(0)} = 0$, choose an integer $r$.
For $k = 1, 2, \ldots$,
    $y^{(0)} = u^{(k-1)}$; $z^{(0)} = v^{(k-1)}$;
    $y^{(j+1)} = \Phi_P(z^{(j)})$, $z^{(j+1)} = \Phi_Q(y^{(j+1)})$, $j = 0, \ldots, r-1$.
    Compute the approximations $t_1^{(r-1)}$ and $t_2^{(r-1)}$ by applying the RRE of Algorithm 1 on
    the vectors $[(y^{(0)}, z^{(0)})^T, (y^{(1)}, z^{(1)})^T, \ldots, (y^{(r)}, z^{(r)})^T]$;
    If $t_1^{(r-1)}$ and $t_2^{(r-1)}$ satisfy accuracy test, stop.
    Else
        $u^{(k)} = t_1^{(r-1)}$; , $v^{(k)} = t_2^{(r-1)}$;
End
```

3.4. The choice of r. The RRE algorithm should be restarted every r iterations, for some integer $r > 1$, to avoid the increase in computational work and storage as k increases.

If $w^* = \Phi(w^*)$ is a fixed point and $J(\Phi, w^*)$ is the Jacobian matrix of Φ at w^*, then

$$w^{(k+1)} - w^* = J(\Phi, w^*)(w^{(k)} - w^*) + O(\|(w^{(k)} - w^*)\|^2).$$

It suffices to examine the eigenvalues of the Jacobian matrix $J(\Phi, w^*)$. Going back to the iteration (2.21) and observing the eigenvalues of the Jacobian matrix (2.25) at the solution, it can be seen that these eigenvalues range between zero and one. Most of these eigenvalues are close or equal to zero, except for a few which are close to one. Therefore, choosing a small integer r is sufficient. In particular, $r = 4$ yields good results. Figure 3.2 shows the distribution of the spectrum of the Jacobian matrix at the solution for $n = 512$, and $(\alpha, c) = (0.001, 0.999)$ with spectral radius $\rho(J(\Phi, (u^*, v^*))) \approx 0.86$.

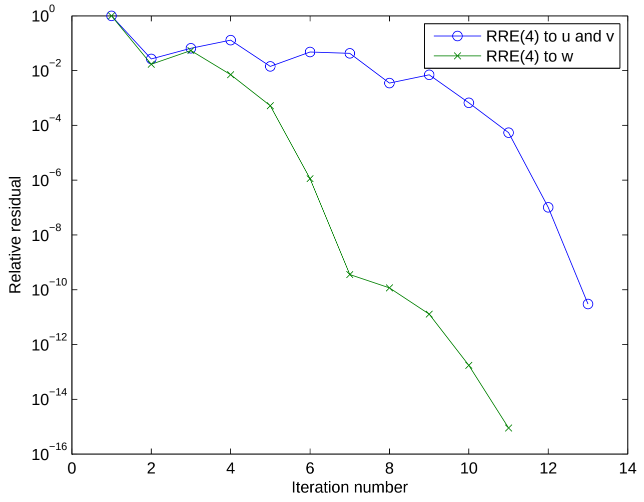

FIG. 3.1. $n = 512,\ \alpha = 0.001,\ c = 0.999$.

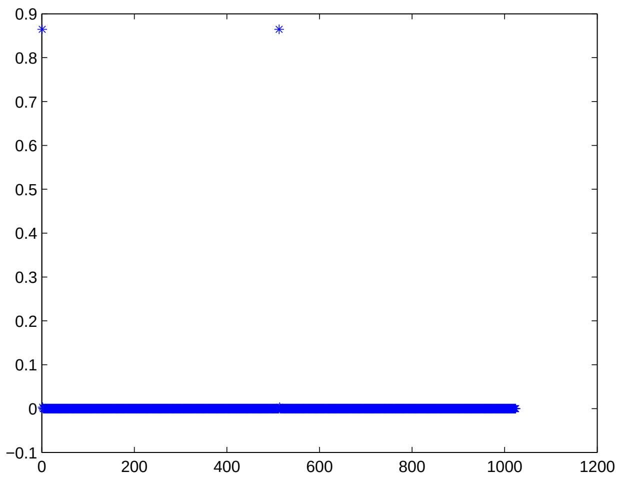

FIG. 3.2. *Distribution of the spectrum of the Jacobian matrix at the solution.*

4. Numerical experiments. A numerical example is presented in this section to illustrate the performance of the new approach for solving the vector equation (2.17). We consider a special kind of Riccati equation (2.3)–(2.5). 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 $n/4$ subintervals of equal length and applying a composite Gauss-Legendre quadrature rule with 4 nodes in each subinterval.

Computations are performed for different choices of the parameters (α, c) and for different values of n using MATLAB 7.4 (R2007a) on an Intel-I5 quad-core 2.5 Ghz processor (4 GB RAM) with Fedora 18 (Linux) and with approximately 15 significant decimal digits.

ETNA
Kent State University
http://etna.math.kent.edu

The stopping criterion is given by

$$ERR = \frac{\|w^{(k+1)} - w^{(k)}\|}{\|w^{(k+1)}\|} \leq tol$$

for $tol = 10^{-10}$.

Table 4.1 compares the three vector extrapolation methods RRE,MPE, and MMPE for two values of r. One can observe that for $r = 4$, the three methods are comparable and give close results, while for a larger r ($r = 10$), the RRE method is the best.

TABLE 4.1
Comparison of the three vector extrapolation methods for different r and for $n = 2048$.

$r = 4$	RRE	MPE	MMPE
CPU time (in seconds)	3.8	3.8	3.9
Residual norm	$3.5.10^{-11}$	2.10^{-11}	$3.9.10^{-11}$

$r = 10$	RRE	MPE	MMPE
CPU time (in seconds)	6.59	7.57	10.36
Residual norm	$6.87.10^{-12}$	9.0^{-12}	$1.0.10^{-11}$

Table 4.2 compares the iterative method proposed by Lin (2.12), its modified version (2.13), and the application of RRE. Denote by ILin the iterative method proposed by Lin [24] with the choice of T_k given in (2.18), by MILin in its modified version, and by RRE the modified version with the application of restarted reduced rank extrapolation every 4 iterations of Algorithm 2. Table 4.2 shows how the three methods converge to the minimal positive solution of (2.20) for several α and c values. The RRE method is seen to outperform ILin and MILin. Figure 4.1 shows the performance of RRE in comparison with ILin and MILin for $(\alpha, c) = (0.001, 0.999)$.

TABLE 4.2
Numerical results for n=256 with different (α,c).

α	c	Method	Iteration steps	Residual norm	CPU time
10^{-8}	$1-10^{-6}$	ILin	4732	9.98e-11	5.47
		MILin	2517	9.98e-11	2.87
		RRE	20	5.03e-14	0.11
10^{-5}	$1-10^{-5}$	ILin	1813	9.97e-11	2.08
		MILin	955	9.93e-11	1.08
		RRE	7	3.83e-14	0.05
0.0001	0.9999	ILin	674	9.98e-11	0.77
		MILin	353	9.88e-11	0.4
		RRE	7	9.8e-16	0.04
0.001	0.999	ILin	246	9.7e-11	0.3
		MILin	129	9.04e-11	0.14
		RRE	9	2.99e-14	0.05
0.5	0.5	ILin	12	4.01e-11	0.01
		MILin	7	3.32e-11	0.008
		RRE	3	9.72e-17	0.02

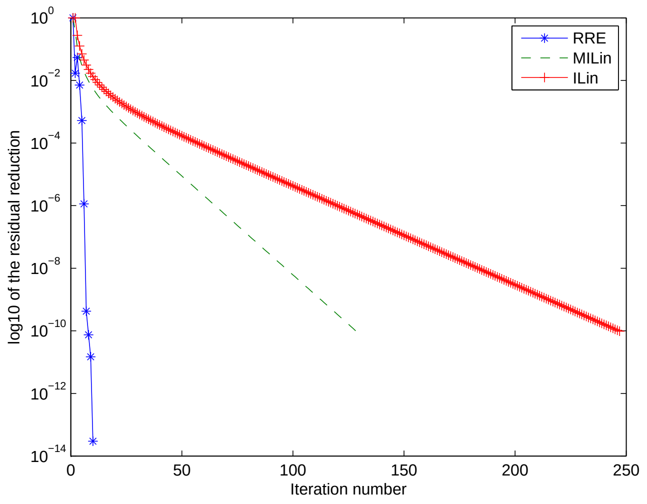

FIG. 4.1. $n = 256,\ r = 4$.

We now compare RRE and the fast Newton method proposed in [4]. Computations of this table have been implemented in Fortran 90 on Linux with 2.5 Ghz and with about 15 significant decimal digits and for $tol = 10^{-10}$. The Fortran code implemented in [4] is used. Denote by LuF the fast Newton method, which is based on a fast LU algorithm that reduces the cost of Newton's algorithm proposed by Lu [26] from $O(n^3)$ to $O(n^2)$; see Section 2.1.3. Table 4.3 compares the restarted RRE method with LuF in terms of CPU time (in seconds) for different n and for $(\alpha,\ c) = (10^{-8}, 1 - 10^{-6})$. It can be seen that RRE is faster than LuF also when the convergence is slow for (α, c) close to $(0, 1)$ and for large n.

TABLE 4.3
Comparison in terms of CPU time in seconds for different n.

$\alpha = 10^{-8},\ c = 1 - 10^{-6}$		
n	LuF	RRE(4)
512	0.35	0.24
1024	1.37	0.96
2048	6.6	3.8

5. Conclusions.

In this paper, we dealt with a special kind of Riccati equations, NARE, that arises in transport theory. We presented some efficient iterative methods which solve this kind of equations either by computing the minimal positive solution of a vector equation or directly to NARE. We applied a polynomial vector extrapolation method, namely reduced rank extrapolation (RRE), to the iterative method proposed by Lin [24] which computes the minimal positive solution of NARE as the minimal positive solution of a vector equation. Numerical experiments showed the advantage of using the RRE method, especially near the critical case when (α, c) is close to $(0, 1)$ and for large n.

Acknowledgments. We would like to thank F. Poloni for providing us with his program for the LuF algorithm. We also thank the referees for their useful comments and valuable suggestions.

REFERENCES

[1] Z. Z. BAI, Y. H. GAO, AND L. Z. LU, *Fast iterative schemes for nonsymmetric algebraic Riccati equations arising from transport theory*, SIAM J. Sci. Comput., 30 (2008), pp. 804–818.

[2] L. BAO, Y. LIN, AND Y. WEI, *A modified simple iterative method for nonsymmetric algebraic Riccati equations arising in transport theory*, Appl. Math. Comput., 181 (2006), pp. 1499–1504.

[3] R. BELLMAN AND G. M. WING, *An Introduction to Invariant Embedding*, John Wiley, New York, 1975.

[4] D. A. BINI, B. IANNAZZO, AND F. POLONI, *A fast Newton's method for a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 276–290.

[5] D. A. BINI, B. MEINI, AND B. IANNAZZO, *Numerical solution of algebraic Riccati equations. Fundamentals of Algorithms*, SIAM, Philadelphia, 2012.

[6] C. BREZINSKI, *Généralisations de la transformation de Shanks, de la table de Padé et de l' ϵ-algorithme*, Calcolo, 12 (1975), pp. 317–360.

[7] C. BREZINSKI AND M. REDIVO ZAGLIA, *Extrapolation Methods. Theory and Practice*, North-Holland, Amsterdam, 1991.

[8] S. CABAY AND L.W. JACKSON, *A polynomial extrapolation method for finding limits and antilimits of vector sequences*, SIAM J. Numer. Anal., 13 (1976), pp. 734–752.

[9] F. CORON, *Computation of the asymptotic states for linear half space kinetic problem*, Transport Theory Statist. Phys., 19 (1990), pp. 89–114.

[10] R. P. EDDY, *Extrapolating to the limit of a vector sequence.*, in Information Linkage between Applied Mathematics and Industry, P. C. C. Wang, ed., Academic Press, New York, 1979, pp. 387–396.

[11] W. F. FORD AND A. SIDI, *Recursive algorithms for vector extrapolation methods* , Appl. Numer. Math., 4 (1988), pp. 477–489 .

[12] B. D. GANAPOL, *An investigation of a simple transport model*, Transport Theory Statist. Phys., 21 (1992), pp. 1–37.

[13] I. GOHBERG, T. KAILATH, AND V. OLSHEVSKY, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1576.

[14] C. H. GUO AND A. J. LAUB, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 376–391.

[15] C. H. GUO AND W. W. LIN, *Convergence rates of some iterative methods for nonsymmetric algebraic Riccati equations arising in transport theory*, Linear Algebra Appl., 432 (2010), pp. 283–291.

[16] K. JBILOU AND H. SADOK, *LU implementation of the modified minimal polynomial extrapolation method for solving linear and nonlinear systems*, IMA J. Numer. Anal., 19 (4) (1999), pp. 549–561.

[17] ———, *Vector extrapolation methods. Applications and numerical comparison*, J. Comput. Appl. Math., 122 (2000), pp. 149–165.

[18] J. JUANG , *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.

[19] J. JUANG, C. L. HSING, AND P. NELSON , *Global existence, asymptotics and uniqueness for the reflection kernel of the angularly shifted transport equation*, Math. Models Methods Appl. Sci., 5 (1995), pp. 239–251.

[20] J. JUANG AND I-DER CHEN, *Iterative solution for a certain class of algebraic matrix Riccati equations arising in transport theory*, Transport Theory Statist. Phys., 22 (1993), pp. 65–80.

[21] J. JUANG AND W. W. LIN, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 228–243.

[22] C. KENNEY AND A. J. LAUB, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.

[23] P. LANCASTER AND L. RODMAN, *Algebraic Riccati equations*, Oxford University Press, New York, 1995.

[24] Y. LIN, *A class of iterative methods for solving nonsymmetric algebraic Riccati equations arising in transport theory*, Comput. Math. Appl., 56 (2008), pp. 3046–3051.

[25] 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), pp. 679–685.

[26] ———, *Newton iterations for a non-symmetric algebraic Riccati equation*, Numer. Linear Algebra Appl., 12 (2005), pp. 191–200.

[27] V. L. MEHRMANN, *The Autonomous Linear Quadratic Control Problem*, Lecture Notes in Control and Information Sciences 163, Springer, Berlin, 1991.

[28] M. MESINA, *Convergence acceleration for the iterative solution of the equations $X = AX + f$*, Comput. Methods Appl. Mech. Engrg., 10 (1977), pp. 165–173.

[29] B. P. PUGATCHEV, *Acceleration of the convergence of iterative processes and a method for solving systems of nonlinear equations,* U.S.S.R. Comput. Math. and Math. Phys., 17 (1977), pp. 199–207.

[30] A. SIDI, *Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms*, SIAM J. Numer. Anal., 23 (1986), pp. 197–209.

[31] ———, *Efficient implementation of minimal polynomial and reduced rank extrapolation methods*, J. Comput.

Appl. Math., 36 (1991), pp. 305–337.

[32] ———, *Extrapolation vs. projection methods for linear systems of equations*, J. Comput. Appl. Math., 22 (1988), pp. 71–88.

[33] A. SIDI, W. F. FORD, AND D. A. SMITH, *Acceleration of convergence of vector sequences,* SIAM J. Numer. Anal., 23 (1986), pp. 178–196.

[34] D. A. SMITH, W. F. FORD, AND A. SIDI, *Extrapolation methods for vector sequences,* SIAM Rev., 29 (1987), pp. 199–233.

[35] P. WYNN, *Acceleration techniques for iterated vector and matrix problems*, Math. Comp., 16 (1962), pp. 301–322.

[36] ———, *On a device for computing the* $e_m(S_n)$ *transformation*, Math. Tables Aids Comput., 10 (1956), pp. 91–96.