U.P.B. Sci. Bull., Series A, Vol.77, Iss. 1, 2015 ISSN 1223-7027

A HYBRID ITERATIVE METHOD FOR A CLASS OF RICCATI EQUATIONS IN THE CRITICAL CASE

Bo Yu[1], Ning Dong*[1], Feng-Hua Wen,[2] Xiao-Hong Chen[2]

In this paper, we devise a hybrid nonlinear block splitting double Newton method to compute the minimal positive solution of a class of Riccati equations arising from transport theory. The overall convergence of our algorithm is established. Preliminary numerical experiments show the new presented method is very efficient for compute the desired solution of equations near or in the critical case.

Keywords: Ricaati equation, nonlinear block splitting, double Newton step, the critical case.
MSC2010: 65F30, 65R20, 65H05, 15A24

1. Introduction

We consider the following nonsymmetric algebraic Riccati equation (NARE) arising from the transport theory [1, 13]

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

where X is the desired solution matrix and A, B, C and $D \in \mathbb{R}^{n\times n}$ are known matrices of forms

$$A = \Delta - eq^T, \quad B = ee^T, \quad C = qq^T, \quad D = \Gamma - qe^T. \tag{2}$$

In the above,

$$\Delta = \mathrm{diag}(\delta_1, \delta_2, ..., \delta_n), \quad \Gamma = \mathrm{diag}(\gamma_1, \gamma_2, ..., \gamma_n),$$
$$e = (1, 1, ..., 1)^T, \quad q = (q_1, q_2, ..., q_n)^T,$$

where

$$\delta_i = \frac{1}{c\omega_i(1+\alpha)}, \quad \gamma_i = \frac{1}{c\omega_i(1-\alpha)}, \quad q_i = \frac{c_i}{2\omega_i}$$

[1]School of Science, Hunan University of Technology, Zhuzhou, 412008, P.R. China. e-mail: boyu_hut@126.com, dongning_158@aliyun.com. (* Corresponding Author)
[2]Business School, Central South University, Changsha, 410083, P.R. China. e-mail: mailtowfh@aliyun.com.

for $i = 1, ..., n$ with constants $\alpha \in [0, 1)$, $c \in (0, 1]$ $c_i > 0$, $w_i > 0$ satisfying

$$0 < \omega_n < ... < \omega_1 < 1, \quad \sum_{i=1}^{n} c_i = 1.$$

From the definition of the diagonal elements of Δ and Γ, we can easily see

$$0 < \delta_1 < \delta_2 < ... < \delta_n \quad \text{and} \quad 0 < \gamma_1 < \gamma_2 < ... < \gamma_n. \tag{3}$$

The above Riccati equation (1) is derived by discretizing the Gauss-Legendre quadrature formula to an integrodifferential equation related to the transport theory [13] or the Nash game [1]. It also can be regarded as a more generalized form of Chandrasekhar H-equation considered in [6, 12, 20]. The minimal positive solution X of (1) is of great interest and its existence was proved by a lot of scholars (see, e.g. [10, 12]). In recent years, various numerical methods have been developed to compute the minimal positive solution (see, e.g. [2]-[11], [18]-[22]). By introducing the M-matrix structure, Guo et.al. [10] first transformed (1) to a more generalized Riccati matrix equation and gave the Newton method in matrix form

$$(A - X^{(k)}C)X^{(k+1)} + X^{(k+1)}(D - CX^{(k)}) = B - X^{(k)}CX^{(k)}, \quad k = 1, 2, ... \tag{4}$$

and the fixed-point method in matrix form

$$A_1X^{(k+1)} + X^{(k+1)}D_1 = X^{(k)}CX^{(k)} + A_2X^{(k)} + X^{(k)}D_2 + B, \quad k = 1, 2, ... \tag{5}$$

to find the minimal positive solution, where in (5) matrices A_1 and A_2, D_1 and D_2 are some regular splitting of coefficient matrices A and D [17].

If the NARE (1) is far away from the critical case, the fixed-point method (5) is more proper to compute the minimal positive solution since its computational cost at each step is relatively cheaper than that of Newton's method although both of them are about $O(n^3)$. While the NARE (1) comes close to the critical case, Newton's method (4) and its double variant (see, e.g. [7]) become a better choice than the fixed-point iteration (5) as they bear a higher convergence speed than that of fixed-point iteration.

As for the computational methods at each step in iterations (4) and (5), the accurate algorithms such as Bartels-Stewart algorithm [5] are enough for small scale problems. When the scale of (1) becomes larger, accurate methods lose their effectiveness as their complexity at each iterative step is considerably huge as n increase. By noting the special structure of NARE (1), Juang [12] observed the solution X is of the form

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

with

$$T := (t_{ij}) = \frac{1}{\delta_i + \gamma_j}, \qquad u = Xq + e, \qquad v = X^T q + e, \tag{6}$$

where $\circ$ is the Hadamard product. Lu [15] made use of these expression to reformulate Ricaati equation to an equivalent form

$$u = u \circ (Pv) + e, \qquad v = v \circ (Qu) + e$$

with

$$P := (P_{ij}) = \frac{q_j}{\delta_i + \gamma_j}, \qquad Q := (Q_{ij}) = \frac{q_j}{\delta_j + \gamma_i} \tag{7}$$

and furthermore, devised the following simple iteration (SI) in vector form

$$\begin{aligned} u^{(k+1)} &= u^{(k)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} &= v^{(k)} \circ (Qu^{(k)}) + e. \end{aligned}$$

The computational superiority of the SI over iterations (4) and (5) is that the complexity at each iterative step can be reduced from $O(n^3)$ to $O(n^2)$, which is fitter to solve (1) with larger scale. Recently, Bai, Gao and Lu [2] further designed a class of nonlinear splitting iteration methods, including the nonlinear block Jacobi (NBJ) iteration

$$\begin{aligned} u^{(k+1)} &= u^{(k+1)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} &= v^{(k+1)} \circ (Qu^{(k)}) + e, \end{aligned}$$

the nonlinear block Gauss-Seidel (NBGS) iteration

$$\begin{aligned} u^{(k+1)} &= u^{(k+1)} \circ (Pv^{(k)}) + e, \\ v^{(k+1)} &= v^{(k+1)} \circ (Qu^{(k+1)}) + e \end{aligned} \tag{8}$$

and the nonlinear block successive overrelaxation (NBSOR) iteration

$$\begin{aligned} u^{(k+1)} \circ (e - Pv^{(k)}) &= se + (1-s)u^{(k)} \circ (e - Pv^{(k)}) \quad (0 < s \le 1), \\ v^{(k+1)} \circ (e - Qu^{(k)}) &= te + (1-t)v^{(k)} \circ (e - Qu^{(k+1)}) \quad (0 < t \le 1). \end{aligned}$$

The most attractive feature of the nonlinear block splitting iterations is that they can obtain faster convergence with less computational complexity compared with that of SI iteration. Especially, the NBGS iteration stands out among all nonlinear block splitting methods as it surpasses others both in CPU time and convergence rate. For other iterative methods with $O(n^2)$ complexity, we refer to [3], [16] for example.

It should be pointed out that although the developed nonlinear block splitting iterations beat the fixed-point iterations for NARE (1) far away from the critical case, they still show the slow sublinear convergence rate when NARE (1) is in the critical case. One approach to tackle this difficulty is employing a shift technique [3] to transfer the NARE (1) to another equation which is no longer in the critical case but shares the same minimal positive solution with (1). In this way one can expect the nonlinear block splitting iterations recover their original convergence speed but they fail to solve the NARE near the critical case, since the solution in the shifted NARE is not the desired one in the original NARE. In this paper, we are bound for another way to enhance the overall convergence of the nonlinear block splitting iterations in or near the critical case by inosculating a double Newton step. The proposed hybrid algorithm mainly depends on two computational switches. One plays the role to detect whether the current nonlinear block splitting iteration needs turning to Newton's iteration and the other can automatically determine a double Newton step is required or not. In another word, these two switches make the algorithm self-adaptive whenever the NARE is in (near) or far away from the critical case. Particularly, Numerical experiments in the last section show that when NARE (1) is near or in the critical case, our algorithm will work very well for computing the minimal positive solution of NARE (1).

The rest of this paper is organized as follows. We propose the hybrid nonlinear block splitting Newton method and construct its overall convergence in Section 2. Section 3 is devoted to describing a double Newton step with the aim to accelerate the new-presented method in the critical case. We do some numerical experiments in Section 4 to indicate the effectiveness of our proposed algorithm.

Notations. Let I_r and I be the identity matrices of order r and n, respectively. For a diagonal matrix $D \in \mathbb{R}^{n\times n}$ and a vector $d \in \mathbb{R}^n$, $\mathrm{diag}(D)$ represents the vector whose elements are the diagonal entries of D and, $\mathrm{diag}(d)$ represents the diagonal matrix whose diagonal entries are elements of d.

2. The hybrid nonlinear block splitting Newton method

In this section, we first give a concise Newton iterative scheme which is equivalent to (4). Then we present the hybrid nonlinear block splitting Newton algorithm to compute the minimal positive solution.

Let $w^T = (u^T, v^T)$. By using the vectors given in (6), it is not difficult to see that the NARE (1) can be rewritten as

$$\mathcal{R}(w) \;=\; \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} \mathrm{diag}(Pv) & 0 \\ 0 & \mathrm{diag}(Qu) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} e \\ e \end{bmatrix} = 0. \qquad (9)$$

By using Newton's method to the equation (9), one can obtain a iterative scheme in vector form

$$\begin{aligned} &\begin{bmatrix} I-\mathrm{diag}(Pv^{(k)}) & -\mathrm{diag}(u^{(k)})P \\ -\mathrm{diag}(v^{(k)})Q & I-\mathrm{diag}(Qu^{(k)}) \end{bmatrix} \begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} \\ = &\begin{bmatrix} 0 & -\mathrm{diag}(u^{(k)})P \\ -\mathrm{diag}(v^{(k)})Q & 0 \end{bmatrix} \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} + \begin{bmatrix} e \\ e \end{bmatrix} \end{aligned} \qquad (10)$$

with matrices P and Q given in (7). As the iterative matrix sequence $\{X^{(k)}\}_{k=0}^{\infty}$ produced from (4) with the initial guess zero matrix is monotonically increasing and convergent to the minimal positive solution X^* (see [10]), we know form (6) that the sequence $\{w^{(k)}\}_{k=0}^{\infty}$ generated by (10) is also monotonically increasing and convergent to w^*, the minimal positive solution of $\mathcal{R}(w) = 0$.

It is worth noting that if computing the desired solution of NARE only by Newton's method, the computational cost at each iteration is relatively higher than that of nonlinear block splitting methods. However when the current iteration point is close enough to the desired solution, Newton's method is more preferred as result of its quadratic convergence. Therefore it is advisable to start with some nonlinear block splitting iteration and then switch to the Newton step provided that the residual error is down to a certain prescribed level. Since the numerical performance of NBGS iteration (8) is the best among the family of nonlinear block splitting iterations [2], we only describe NBGS-Newton algorithm as follows and other hybrid methods can be derived in a similar manner.

Algorithm 2.1.
1. Set $w^{(0)} = 0$.
2. For $k = 1, 2, ...k_0$,
   compute $w^{(k+1)}$ via NBGS iteration (8);
3. For $k = k_0, k_0 + 1, ...$ until convergence,
   solve (10) to obtain $w^{(k+1)}$.

To show the overall convergence in step 3 of Algorithm 2.1, we first take a second to recall the monotonic convergence of NBGS iteration [2] and Newton's iteration [16].

Lemma 1. *Let the sequence $\{w^{(k)}\}_{k=0}^{\infty}$ be produced by NBGS iteration (8) with $w^{(0)} = 0$. Then for $k = 1, 2, ...$, it holds that*

$$0 \leq w^{(k)} < w^{(k+1)} < w^*$$

and $\lim_{k\to\infty} w^{(k)} = w^*$.

Lemma 2. *Let the sequence $\{w^{(k)}\}_{k=0}^{\infty}$ be produced by Newton's iteration (10). Then for $k = 1, 2, ...$, it holds that*

$$0 \leq w^{(k)} < w^{(k+1)} < w^*$$

and $\lim_{k\to\infty} w^{(k)} = w^*$.

We now establish the overall convergence of Algorithm 2.1.

Theorem 3. *If $\{(w^{(k)}\}_{k=0}^{k_0}$ is produced by NBGS iteration (8) with $w^{(0)} = 0$ and $\{w^{(k)}\}_{k=k_0+1}^{\infty}$ is generated by Newton's method (10) with $w^{(k_0)}$ as an initial point, then*

$$0 < w^{(1)} < w^{(2)} < ... < w^{(k_0)} \leq w^{(k_0+1)} < ...,$$

and $\lim_{k\to\infty} w^{(k)} = w^*$.

Proof. It follows from Lemma 1 that the sequence $\{w^{(k)}\}_{k=0}^{k_0}$ satisfies

$$0 \leq w^{(1)} < w^{(2)} < ... < w^{(k_0)} < w^*.$$

Moreover for $1 \leq k \leq k_0$, by recalling the NBGS iteration scheme (8), NARE (9) at $w^{(k)}$ admits

$$\begin{aligned}
\mathcal{R}(w^{(k)}) &= \begin{bmatrix} I - \mathrm{diag}(Pv^{(k)}) & 0 \\ 0 & I - \mathrm{diag}(Qu^{(k)}) \end{bmatrix} \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} - \begin{bmatrix} e \\ e \end{bmatrix} \\
&= \begin{bmatrix} I - \mathrm{diag}(Pv^{(k)}) & 0 \\ 0 & I - \mathrm{diag}(Qu^{(k)}) \end{bmatrix} \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} \\
&\quad - \begin{bmatrix} I - \mathrm{diag}(Pv^{(k-1)}) & 0 \\ 0 & I - \mathrm{diag}(Qu^{(k)}) \end{bmatrix} \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} \\
&= \begin{bmatrix} -(P(v^{(k)} - v^{(k-1)})) \circ u^{(k)} \\ 0 \end{bmatrix} \leq 0. \qquad (11)
\end{aligned}$$

Now the Newton's method from the k_0-th step yields

$$\mathcal{R}'(w^{(k_0)})(w^{(k_0+1)} - w^{(k_0)}) = -\mathcal{R}(w^{(k_0)}) \geq 0,$$

which is equivalent to

$$\begin{bmatrix} I - \mathrm{diag}(Pv^{(k_0)}) & -\mathrm{diag}(u^{(k_0)})P \\ -\mathrm{diag}(v^{(k_0)})Q & I - \mathrm{diag}(Qu^{(k_0)}) \end{bmatrix} \begin{bmatrix} u^{(k_0+1)} - u^{(k_0)} \\ v^{(k_0+1)} - v^{(k_0)} \end{bmatrix} \geq 0. \qquad (12)$$

On the other hand, it follows from [10] that the matrix

$$\begin{bmatrix} I - \mathrm{diag}(Pv^*) & -\mathrm{diag}(u^*)P \\ -\mathrm{diag}(v^*)Q & I - \mathrm{diag}(Qu^*) \end{bmatrix}$$

is a nonsingular M-matrix or a singular irreducible M-matrix. Subsequently, the matrix on left hand side of (12) must be a nonsingular M-matrix and all elements of its inverse matrix are greater than or equal zero. Therefore, we have $w^{(k_0+1)} \geq w^{(k_0)}$. Once such a fact holds true, we know from Lemma 2 that

$$w^{(k_0)} \leq w^{(k_0+1)} < ...$$

and $\lim_{k\to\infty} w^{(k)} = w^*$. □

The above theorem shows that the iterative sequence $\{w^{(k)}\}_{k=1}^{\infty}$ generated by Algorithm 2.1 converges to the minimal positive solution of (9). But when NARE (9) is in the critical case, Newton's method will take on linear convergence which results in more iterations in Algorithm 2.1. To accelerate the current Newton iteration, we will give a double Newton step as stated in next section.

3. Double Newton step for NARE in the critical case

When NARE (1) is in the critical case, i.e. the Frechet derivative $\mathcal{R}'(w^*)$ at w^* is singular, the convergence of Newton's method become linear with a constant $1/2$. In this case, a double Newton step can be employed to speed up the convergence. We first show an useful inequality referred as Banach lemma [14].

Lemma 4. *If A and B are $n \times n$ matrices and B is an approximate inverse of A (i.e. $||I - BA|| < 1$), then A and B are both nonsingular and*

$$||A^{-1}|| \le \frac{||B||}{1 - ||I - BA||}.$$

Theorem 5. *Let w^* be the minimal positive solution of $\mathcal{R}(w) = 0$. Suppose that $\mathcal{R}'(w^*)$ is singular. Let $\mathcal{N} = Ker(\mathcal{R}'(w^*))$ and $\mathcal{X} = Im(\mathcal{R}'(w^*))$ be the null space and the range of $\mathcal{R}'(w^*)$, respectively. Let $P_{\mathcal{N}}$ and $P_{\mathcal{X}}$ be the projection on the null space $\mathcal{N}$ and the range $\mathcal{X}$. Assume $\mathbb{R}^n = \mathcal{N} \oplus \mathcal{M}$ with $\oplus$ denoting the direct sum. Let $\{w^{(k)}\}_{k=k_0}^{\infty}$ be generated by Newton's method (10).*

(i) If for $k \ge k_0$, $w^{(k)} - w^* \in \mathcal{N}$, then we have

$$w^{(k+1)} - w^* = \frac{1}{2}(w^{(k)} - w^*), \tag{13}$$

$$\mathcal{R}(w^{(k+1)}) = \frac{1}{4}\mathcal{R}(w^{(k)}). \tag{14}$$

(ii) Assume for $k \ge 1$

$$||(\mathcal{R}'(w^{(k)}))^{-1}|| \le c_1 ||w^{(k)} - w^*||^{-1}. \tag{15}$$

If for sufficiently small $\epsilon > 0$,

$$||P_{\mathcal{X}}(w^{(k)} - w^*)|| < \epsilon ||P_{\mathcal{N}}(w^{(k)} - w^*)||,$$

then

$$||w^{(k)} - 2(\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}(w^{(k)}) - w^*|| \le c\epsilon \tag{16}$$

with some constant c independent of k and ϵ.

Proof. (i) Let $\bar{w}^{(k)} = w^{(k)} - w^*$. Note $\mathcal{R}'(w^*)(\bar{w}^{(k)}) = 0$ as $\bar{w}^{(k)} \in \mathcal{N}$, we have the expansion

$$\begin{aligned} \mathcal{R}'(w^{(k)})(\bar{w}^{(k)}) &= \mathcal{R}'(w^*)(\bar{w}^{(k)}) + \mathcal{R}''(w^*)(\bar{w}^{(k)}, \bar{w}^{(k)}) \\ &= 2\Big(\mathcal{R}(w^*) + \mathcal{R}'(w^*)(\bar{w}^{(k)}) + \frac{1}{2}\mathcal{R}''(w^*)(\bar{w}^{(k)}, \bar{w}^{(k)})\Big) \\ &= 2\mathcal{R}(w^{(k)}). \end{aligned}$$

Then

$$\bar{w}^{(k+1)} = \bar{w}^{(k)} - (\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}(w^{(k)}) = \frac{1}{2}\bar{w}^{(k)}, \tag{17}$$

that is to say the equality (13) holds true. On the other hand, by (17) we have the expansion of $\mathcal{R}(w^{(k+1)})$ at w^*

$$\begin{aligned}\mathcal{R}(w^{(k+1)}) &= \mathcal{R}'(w^*) + \mathcal{R}'(w^*)(\bar{w}^{(k+1)}) + \frac{1}{2}\mathcal{R}''(w^*)(\bar{w}^{(k+1)}, \bar{w}^{(k+1)}) \\ &= \frac{1}{4}\Big(\mathcal{R}'(w^*) + \mathcal{R}'(w^*)(\bar{w}^{(k)}) + \frac{1}{2}\mathcal{R}''(w^*)(\bar{w}^{(k)}, \bar{w}^{(k)})\Big) \\ &= \frac{1}{4}\mathcal{R}(w^{(k)}),\end{aligned}$$

which means the equality in (14) is true.

(ii) Let $\hat{w}^{(k)} = w^* + P_{\mathcal{N}}(w^{(k)} - w^*)$. There must be a constant $c_2 > 0$ such that

$$\begin{aligned}||w^{(k)} - \hat{w}^{(k)}|| &= ||P_{\mathcal{X}}(w^{(k)} - w^*)|| \\ &< \epsilon||P_{\mathcal{N}}(w^{(k)} - w^*)|| \\ &\leq c_2\epsilon||w^{(k)} - w^*||. \end{aligned} \tag{18}$$

Then we have

$$\begin{aligned}& ||I - (\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}'(\hat{w}^{(k)})|| \\ \leq\; & ||(\mathcal{R}'(w^{(k)}))^{-1}|| \cdot ||\mathcal{R}'(w^{(k)}) - \mathcal{R}'(\hat{w}^{(k)})|| \\ \leq\; & c_1||w^{(k)} - w^*||^{-1}c_2||w^{(k)} - \hat{w}^{(k)}|| \\ \leq\; & c_3\epsilon \end{aligned} \tag{19}$$

with $c_3 = c_1c_2$. Therefore, it follows from Lemma 4 that $\mathcal{R}'(\hat{w}^{(k)})$ is nonsingular and

$$\begin{aligned}||(\mathcal{R}'(\hat{w}^{(k)}))^{-1}|| &\leq \frac{1}{1 - c_3\epsilon}||(\mathcal{R}'(w^{(k)}))^{-1}|| \\ &\leq c_4||w^{(k)} - w^*||^{-1} \end{aligned} \tag{20}$$

with some constant $c_4 > 0$, where the second inequality is valid because of (15).

On the other hand, by (18) we can find positive constants c_5 and c_6 such that

$$\begin{aligned}||\mathcal{R}(\hat{w}^{(k)})|| &= ||\mathcal{R}(\hat{w}^{(k)}) - \mathcal{R}(w^*)|| \\ &\leq c_5||\hat{w}^{(k)} - w^*|| \\ &\leq c_5(c_2\epsilon + 1)||w^{(k)} - w^*|| \end{aligned} \tag{21}$$

and

$$\begin{aligned}||\mathcal{R}(\hat{w}^{(k)}) - \mathcal{R}(w^{(k)})|| &\leq c_6||\hat{w}^{(k)} - w^{(k)}|| \\ &\leq c_6c_2\epsilon||w^{(k)} - w^*||. \end{aligned} \tag{22}$$

Therefore we have the following estimate for the factorization

$$
\begin{aligned}
& ||(\mathcal{R}'(\hat{w}^{(k)}))^{-1}\mathcal{R}(\hat{w}^{(k)}) - (\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}(w^{(k)})|| \\
\leq\ & ||((\mathcal{R}'(\hat{w}^{(k)}))^{-1} - (\mathcal{R}'(w^{(k)}))^{-1})\mathcal{R}(\hat{w}^{(k)})|| \\
& +||(\mathcal{R}'(w^{(k)}))^{-1}(\mathcal{R}(\hat{w}^{(k)}) - \mathcal{R}(w^{(k)}))|| \\
\leq\ & ||(\mathcal{R}'(\hat{w}^{(k)}))^{-1}|| \cdot ||\mathcal{R}'(\hat{w}^{(k)})(\mathcal{R}'(w^{(k)}))^{-1} - I|| \cdot ||\mathcal{R}(\hat{w}^{(k)})|| \\
& +||(\mathcal{R}'(w^{(k)}))^{-1}|| \cdot ||\mathcal{R}(\hat{w}^{(k)}) - \mathcal{R}(w^{(k)})|| \\
\leq\ & \bar{c}\epsilon
\end{aligned}
$$

with some $\bar{c} \geq (c_2\epsilon + 1)c_3c_4c_5 + c_1c_2c_6$, where the last inequality holds true because of (15), (19), (20), (21) and (22).

At last by (18) and

$$
\begin{aligned}
& ||w^{(k)} - 2(\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}(w^{(k)}) - w^*|| \\
\leq\ & ||\hat{w}^{(k)} - w^{(k)}|| \\
& +||\hat{w}^{(k)} - w^* - 2((\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}(w^{(k)}))|| \\
\leq\ & ||\hat{w}^{(k)} - w^{(k)}|| \\
& +2||(\mathcal{R}'(\hat{w}^{(k)}))^{-1}\mathcal{R}(\hat{w}^{(k)}) - (\mathcal{R}'(w^{(k)}))^{-1}\mathcal{R}(w^{(k)})||,
\end{aligned}
$$

(16) holds true readily. □

We know from Theorem 5 (i) that the convergence of Newton's method will degrade to linearity with a constant 1/2 when $\mathcal{R}'(w^*)$ is singular. Fortunately in such case, inequality (16) in Theorem 5 (ii) implies that a double Newton step can make the current iteration point remarkable close to the desired solution, which becomes the motivation for the acceleration of the Newton's method by imposing a double step. We describe the overall computational details in Algorithm 3.1 as below.

Algorithm 3.1.

1. Choose parameters k_0, ϵ, $\eta_1 > 0$ and $\eta_2 > 0$.
2. Set $w^{(0)} = 0$, $\mathcal{R}(w^{(0)}) = e_{2n}^T$, $r_0 = ||\mathcal{R}(w^{(0)})||$.
3. For $k = 0, 1, 2, ...,$ do:
    - solve (8) to obtain $w^{(k+1)}$;
    - compute $\mathcal{R}(w^{(k)})$, $r_k = ||\mathcal{R}(w^{(k)})||$;
    - if $r_k/r_0 < \eta_1$ or $k \geq k_0$, goto step 4;
    - update current point $w^{(k)}$.
4. For $p = k, k+1, ...,$ do:
    - solve (10) to obtain $w^{(p+1)}$;
    - compute $\mathcal{R}(w^{(p+1)})$, $r_{p+1} = ||\mathcal{R}(w^{(p+1)})||$;
    - if $r_{p+1}/r_0 < \epsilon$, then stop and $w^* \approx w^{(p+1)}$;

TABLE 4.1 *Test Results for* $(\alpha, c) = (1e-10, 1-(1e-10))$.

n	Method	NBGS	NEWTON	ALG.3.1
64	CPU	0.7344	0.1094	0.0781
	IT	10000/0/0	0/19/0	273/10/0
	RES	7.33e-09	1.99e-15	3.99e-15
128	CPU	2.4688	0.3594	0.2969
	IT	10000/0/0	0/19/0	273/10/0
	RES	2.22e-09	1.11e-15	1.11e-15
256	CPU	8.6563	3.5625	1.5156
	IT	10000/0/0	0/19/0	273/10/0
	RES	7.35e-09	2.88e-15	6.21e-15
512	CPU	29.453	25.250	14.935
	IT	10000/0/0	0/18/0	274/10/0
	RES	7.36e-09	7.54e-15	3.68e-14

if $|\frac{r_{p+1}}{r_p} - \frac{1}{4}| < \eta_2$, then $w^{(p+1)} = w^{(p)} - 2(\mathcal{R}'(w^{(p)}))^{-1}\mathcal{R}(w^{(p)})$
and $r = ||\mathcal{R}'(w^{(p+1)})||$;
if $r/r_0 < \epsilon$, then stop and $w^* \approx w^{(p+1)}$.

Remark. The parameter k_0 is to set the maximal number of the NBGS iteration while η_1 and η_2 are employed to give the error tolerance of the NBGS iteration and the double Newton iteration, respectively.

4. Numerical examples

In this section, we test the effectiveness of the proposed Algorithm 3.1 for NARE (1) with various dimension n. The constants c_i and w_i in NARE 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 Gauss-Legendre quadrature with 4 nodes to each subinterval. We coded Algorithm 3.1 in MATLAB 7.1 with $k_0 = 500$, $\eta_1 = 10^{-5}$ and $\eta_2 = 10^{-6}$. We compared the performances of Algorithm 3.1 with that of NBGS iteration and the Newton's iteration for problems near or in the critical case with $n = 64, 128, 256, 512$.

The obtained results are listed in Table 4.1-4.4 where the "n" column gives the sizes of the problem, the "CPU" row denotes the CPU time used in seconds, the "IT" row represents "the maximal NBGS iteration/ the maximal Newton iteration/ the maximal double Newton iteration". The "RES" row reports the relative residual error

$$\text{RES} = ||\mathcal{R}(w^{(k)})||_\infty / ||\mathcal{R}(w^{(0)})||_\infty,$$

where $w^{(k)}$ is the obtained approximative solution.

We see from Table 4.1-4.3 that, for NARE (1) near the critical case, the NBGS

TABLE 4.2 *Test Results for* $(\alpha, c) = (1e - 13, 1 - (1e - 13))$.

n	Method	NBGS	NEWTON	ALG.3.1
64	CPU	0.7188	0.0938	0.0938
	IT	10000/0/0	0/23/0	273/16/0
	RES	7.46e-09	2.22e-15	3.10e-15
128	CPU	2.4375	0.4219	0.4063
	IT	10000/0/0	0/22/0	274/16/0
	RES	7.48e-09	7.10e-15	3.77e-15
256	CPU	8.640	3.4844	2.6404
	IT	10000/0/0	0/22/0	274/15/0
	RES	7.49e-09	7.99e-15	4.97e-14
512	CPU	32.109	28.765	18.421
	IT	10000/0/0	0/22/0	274/15/0
	RES	7.49e-09	1.21e-14	5.01e-14

TABLE 4.3 *Test Results for* $(\alpha, c) = (1e - 15, 1 - (1e - 15))$.

n	Method	NBGS	NEWTON	ALG.3.1
64	CPU	0.7969	0.0938	0.0469
	IT	10000/0/0	0/24/0	273/5/1
	RES	7.46e-09	4.21e-15	4.21e-15
128	CPU	2.3906	0.4688	0.2188
	IT	10000/0/0	0/24/0	273/5/1
	RES	7.48e-09	3.10e-15	4.66e-15
256	CPU	8.2656	3.5469	1.500
	IT	10000/0/0	0/24/0	273/5/1
	RES	7.49e-09	1.28e-14	7.77e-15
512	CPU	31.812	29.562	11.062
	IT	10000/0/0	0/23/0	274/5/1
	RES	7.49e-09	1.31e-14	9.10e-15

TABLE 4.4 *Test Results for* $(\alpha, c) = (0, 1)$.

n	Method	NBGS	NEWTON	ALG.3.1
64	CPU	0.7344	0.0938	0.0313
	IT	10000/0/0	0/24/0	273/5/1
	RES	7.46e-09	3.77e-15	1.77e-15
128	CPU	2.4219	0.4219	0.2188
	IT	10000/0/0	0/24/0	273/5/1
	RES	7.48e-09	5.10e-15	2.66e-15
256	CPU	8.6563	3.5625	1.5156
	IT	10000/0/0	0/23/0	273/5/1
	RES	7.49e-09	1.37e-14	5.99e-15
512	CPU	31.718	29.818	10.968
	IT	10000/0/0	0/23/0	274/5/1
	RES	7.49e-09	1.28e-14	8.21e-15

iteration fail to attain the prescribed accuracy within 10000 iterations and the proposed Algorithm 3.1 outperforms the Newton's method in CPU time. Especially for NARE (1) is very close to the critical case, numerical results in Table 4.3 indicate Algorithm 3.1 beats other two algorithms both in CPU time and the relative residual error. When NARE (1) is in the critical case, Table 4.4 also shows a similar result to that of Table 4.3, which means the proposed Algorithm 3.1 is very efficient for solving NARE (1) near or in the critical case.

5. Conclusion

We have presented a hybrid nonlinear block splitting double Newton method to compute the minimal positive solution for a class of nonsymmetric algebraic Riccati equations in the critical case. We also constructed the overall convergence of the hybrid algorithm under mild conditions. Numerical experiments particularly indicated that our algorithm is very effective for computing the solution of nonsymmetric algebraic Riccati equations near or in the critical case.

Acknowledgement. The authors are grateful for the anonymous referees' helpful comments. This work was supported partly by the NSF of China (11301170, 71171024, 70921001), Foundation of Education Department of Hunan Province (12C0075, 13C037).

R E F E R E N C E S

[1] *T. P. Azevedo-Perdicoulis and G. Jank*, Iterative solution of algebraic matrix Riccati equations in open loop Nash games, Internat. J. Robust Nonlinear Control, **15** (2005), pp. 55–62.

[2] *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.

[3] *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.

[4] *D. A. Bini, B. Meini and F. Poloni*, Fast solution of a certain Riccati equation through Cauchy-like matrices, Electron. Trans. Numer. Anal., **33** (2009), pp. 84–104.

[5] *R. H. Bartels and G. W. Stewart*, Solution of the matrix equation $AX + XB = C$, Comm. ACM, **15** (1972), pp. 820-826.

[6] *S. Chandrasekhar*, Radiative transfer, Dover, New York, 1960.

[7] *D. W. Decker and C. T. Kelley*, Convergence acceleration for Newton's method at singular points, SIAM J. Numer. Anal., **19** (1982) pp. 219-229.

[8] *Y.-H Gao and Z.-Z Bai*, On inexact Newton methods based on doubling iteration scheme for nonsymmetric algebraic Riccati equations, Numer. Linear Algebra Appl., **18** (2011), pp. 325–341.

[9] *C.-H. Guo*, Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices, SIAM J. Matrix Anal. Appl., **23** (2001), pp. 225–242.

[10] *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.

[11] *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.

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

[13] *J. Juang and W.-W. Lin*, Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices, SIAM J. Matrix Anal. Appl. **20** (1998), pp. 228–243.

[14] *C. T. Kelley*, Iterative methods for linear and nonlinear equations, Society for lndustrial and Applied Mathematics, Philadelphia, (1995).

[15] *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.

[16] *L.-Z. Lu*, Newton iterations for a nonsymmetric algebraic Riccati equation, Numer. Linear Algebra Appl., **12** (2005), pp. 191–200.

[17] *R. Varga*, Matrix Iterative Analysis, (2nd Edition), Springer-Verlag Berlin Heidelberg, (2000).

[18] *B. Yu and N. Dong*, A Structure-preserving doubling algorithm for quadratic matrix equations arising form damped mass-spring system, Advan. Model. Optimiz., **12** (2010), pp. 85-100.

[19] *B. Yu, N. Dong, Q. Tang and F.-H. Wen*, On iterative methods for the quadratic matrix equation with M-matrix. Appl. Math. Comp., **218** (2011), pp. 3303-3310.

[20] *B. Yu and N. Dong*, A Structured Newton's method for Chandrasekhar H-equation. University Politehnica of Bucharest Scientific Bulletin -Series A- Applied Mathematics and Physics, **73** (2011), pp. 29–44.

[21] *B. Yu, D.-H. Li and N. Dong*, Convergence of the cyclic reduction algorithm for a class of weakly overdamped quadratics. J. Comp. Math., **30** (2012), pp. 139-156.

[22] *B. Yu, D.-H. Li and N. Dong*, Lowmemory and low complexity iterative schemes for a nonsymmetric algebraic Riccati equation arising from transport theory. J. Comput. Appl. Math., **250** (2013), 175-189.