

# A STRUCTURED NEWTON'S METHOD FOR CHANDRASEKHAR H-EQUATION ARISING FROM RADIATIVE TRANSFER

Bo YU,<sup>1 2</sup> Ning DONG<sup>2</sup>

*By exploiting the implicit structure of Chandrasekhar H-equation, we devise a structured Newton's method to compute the solution with physical meaning. We also consider the critical case of this equation, in which the Jacobian matrix at the solution is singular. By incorporating a shift technique, we show that our structured Newton's method can still converge quadratically to the solution. Numerical experiments are listed to indicate that the new method is a feasible and effective solver for Chandrasekhar H-equation.*

**Keywords:** Chandrasekhar H-equation, Newton's method, factored-alternating-directional-implicit iteration, shift technique.

**MSC2000:** 45G05, 45G10, 65R20, 65H05

## 1. Introduction

The Chandrasekhar H-equation

$$F(H)(\delta) = H(\delta) - \left(1 - \frac{c}{2} \int_0^1 \frac{\delta H(t) dt}{\delta + t}\right)^{-1} = 0 \quad (1)$$

arises from solving exit distribution problems in radiative transfer [4, 17, 18]. Here the parameter  $c$  is the average total number of particles emerging from a collision, which is assumed to be nonnegative and conservative, i.e.  $0 \leq c \leq 1$ . By the use of some numerical integral formula such as composite midpoint rule

$$\int_0^1 f(\delta) d\delta \approx \frac{1}{n} \sum_{j=1}^n f(\delta_j) \quad (2)$$

with  $\delta_i = (i - 1/2)/n$  for  $1 \leq i \leq n$ , the integral equation (1) can be transformed into a nonlinear vector equation

$$F(x) = 0 \quad (3)$$

---

<sup>1</sup>College of Mathematics and Econometrics, Hunan University, Changsha, 410082, P. R. China. e-mail: wenyubwenyub@yahoo.com.cn

<sup>2</sup>School of Science, Hunan University of Technology, Zhuzhou, 412008, P.R. China. e-mail: dnsx216@tom.com.

with its  $i$ -th element

$$F(x)_i = x_i - \left(1 - \frac{c}{2n} \sum_{j=1}^n \frac{\delta_i x_j}{\delta_i + \delta_j}\right)^{-1} = 0.$$

When  $c \in (0, 1)$ , the H-equation has exactly two positive solutions [13, 18], but only the minimal positive solution is of great interest in physics. Various iteration methods for finding the minimal positive solution have been studied. In [16], the author proposed a fixed-point iteration method for the coupled system of H-equations and proved its convergence. Newton's method and the chord method, which possess quadratical and linear convergence, were presented in [17]. In [18], Krylov subspace methods such as GMRE [24], BiCGSTAB [26] and TRQMR [7] were employed to solve the subproblem of Newton's method. For more discussion of other methods for H-equation, see [5, 6, 19, 20] and its references therein. These methods are effective solvers for H-equation but a common feature of them is that the implicit structure in (3) is not explored sufficiently.

It has been shown in [17] that an equivalent form of H-equation (1) is

$$G(H)(\delta) = H(\delta) - 1 - \frac{c}{2} H(\delta) \int_0^1 \frac{\delta H(t) dt}{\delta + t} = 0,$$

and its discrete equation, by using the same numerical integral rule as in (2), takes the form

$$G(x) = x - x \circ Sx - e = 0, \quad (4)$$

where  $e^T = (1, \dots, 1)$  and  $S$  is a matrix with

$$(S)_{ij} = \alpha \frac{\delta_i}{\delta_i + \delta_j} \quad \text{and} \quad \alpha = \frac{c}{2n}. \quad (5)$$

Applying Newton's method directly to H-equation (4) yields the following iteration scheme

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - M_k^{-1}(x^{(k)} - x^{(k)} \circ Sx^{(k)} - e) \\ &= M_k^{-1}(e - x^{(k)} \circ Sx^{(k)}), \quad k = 0, 1, \dots, \end{aligned} \quad (6)$$

with  $M_k = I - \text{diag}(Sx^{(k)}) - \text{diag}(x^{(k)})S$  and “ $\circ$ ” denoting the Hadamard product symbol. However, this Newton's method does not differ very much from the original one since what we have done is only giving a small change in representation of variables and the structure is not unfolded yet.

The purpose of this paper is to provide a structured Newton's method to obtain the minimal positive solution of H-equation (4). To this purpose, we define an unknown Cauchy-like matrix  $X$  (introduced in the next section) so that the H-equation (4) can be written as a Riccati matrix equation

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

with  $A$ ,  $B$  and  $C$  some known coefficient matrices of dimension  $n \times n$ . Newton's method with an initial guess for this Riccati equation is given by

$$(A - X^{(k)}C)X^{(k+1)} + X^{(k+1)}(A^T - CX^{(k)}) = B - X^{(k)}CX^{(k)}, \quad k = 0, 1, \dots \quad (7)$$

Obviously, if there is no any special structure in (7), the complexity of Newton's method in the matrix form is definitely higher than that of Newton's method (6) in the vector form. Fortunately, we observe that some structured characteristics lie in the equation (7) as follows:

- (i) . The right hand side of the equation (7) is a low rank matrix.
- (ii) . The coefficient matrices in left hand side of (7) are diagonal plus rank one matrices and their eigenvalues are positive real numbers and can be evaluated in  $O(n)$  flops.

These attractive structures allow us to design a structured Newton's method to obtain the minimal positive solution of H-equation (4). The idea is to apply a factored-alternating-directional-implicit (FADI) method [3, 21], motivated by the structure feature (i), to solve the subproblem equation (7) to get  $X^{(k+1)}$ , then form the the next iterative vector  $x^{(k+1)}$  from  $X^{(k+1)}$ . This strategy is very efficient because the structure feature (ii) guarantees that the optimal ADI parameters can be computed cheaply through fast evaluating the two extremal eigenvalues of coefficient matrices (see Wachspress's method [27, 28]). Moreover, it is unnecessary that the iterative matrix  $X^{(k+1)}$  should be formed exactly since only a matrix-vector product is needed in each Newton iteration step. Therefore, we can expect that such structured Newton's method may be implemented in a low storage way to obtain the minimal positive solution of H-equation (4).

Another important case for H-equation is the critical case, i.e.  $c = 1$  in (1). The H-equation in this case has a unique positive solution [13]. Also, the Jacobian at this solution is singular, which consequently causes that Newton's method is convergent linearly. This drawback can be removed by incorporating a shift technique, introduced by He, Meini and Rhee [12] (see also [2, 10]), into our transformed Riccati equation. Hence, the proposed structured Newton's method can still be employed to obtain the positive solution of H-equation (4) with quadratical convergence and high accuracy.

The rest of this paper is organized as follows. We give some preliminaries in Section 2. In Section 3, we describe the structured Newton's method for H-equation with  $c \in (0, 1)$ . Section 4 is devoted to the H-equation when  $c = 1$ , where the Jacobian of Newton's method at the solution is singular. At last, we do some numerical experiments in Section 5 to test the proposed methods and compare their performances with those of Newton-GMRE, Newton-BiCGSTAB and Newton-TRQMR in [18].

Throughout this paper, we use " $\otimes$ " to denote Kronecker product of matrices. Let  $I$  be the identity matrix of order  $n$  and  $\mathcal{J} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$ . For matrix  $A \in \mathbb{R}^{n \times n}$ ,

we denote by  $\sigma(A)$  and  $\rho(A)$  its spectrum and spectral radius, respectively. For a diagonal matrix  $D \in \mathbb{R}^{n \times n}$  and a vector  $d \in \mathbb{R}^n$ ,  $\text{diag}(D)$  represents the vector whose elements are the diagonal entries of  $D$  and,  $\text{diag}(d)$  represents the diagonal matrix whose diagonal entries are elements of  $d$ .

## 2. Preliminaries

We give some preliminaries in this section. The first concept is the Cauchy-like matrix [15]. An  $n \times n$  matrix  $X$  is called real Cauchy-like matrix if its entries have the form

$$(X)_{ij} = \frac{f_i g_j}{\delta_i + \gamma_j},$$

where  $f_i, g_i, \delta_i$  and  $\gamma_i$  ( $i = 1, \dots, n$ ) denote some scalars such that  $\delta_i + \gamma_j \neq 0$  for any  $i \neq j$ . The Cauchy-like matrix  $X$  satisfies the following displacement equation

$$\Delta X + X \Gamma = f g^T$$

with diagonals  $\Delta = \text{diag}(\delta_1, \dots, \delta_n)$ ,  $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_n)$  and vectors  $f^T = (f_1, \dots, f_n)$ ,  $g^T = (g_1, \dots, g_n)$ . We also need the concept of M-matrix. A real square matrix  $A$  is called a Z-matrix if all its off-diagonal elements are nonpositive. It is clear that a Z-matrix  $A$  can be written as  $sI - B$  with  $(B)_{ij} \geq 0$ . A Z-matrix  $sI - B$  with  $(B)_{ij} \geq 0$  is called a singular or nonsingular M-matrix if  $s = \rho(B)$  or  $s > \rho(B)$ .

The following lemma comes from [25] which gives some interesting properties of the M-matrix.

**Lemma 1.** *For a Z-matrix  $A$ , it holds true that*

- (a)  *$A$  is an M-matrix if and only if  $u^T A \geq 0$  ( $Av \geq 0$ ) for some vector  $u \geq 0$  ( $v \geq 0$ );*
- (b) *when  $A$  is nonsingular,  $A$  is an M-matrix if and only if  $A^{-1} \geq 0$  or,  $u^T A > 0$  ( $Av > 0$ ) for some vector  $u > 0$  ( $v > 0$ ) or,  $\sigma(A) \in \mathbb{C}^+$  with  $\mathbb{C}^+$  denoting the open right half-plane.*

The following lists some well-known results about a nonsymmetric algebraic Riccati equation structure

$$R(Y) = Y\bar{C}Y - \bar{A}Y - Y\bar{D} + \bar{B} = 0, \quad (8)$$

see [9, 10, 11].

**Theorem 2.** *Let*

$$\bar{\mathcal{M}} = \begin{bmatrix} \bar{D} & -\bar{C} \\ -\bar{B} & \bar{A} \end{bmatrix} \quad (9)$$

*be a nonsingular or irreducible singular M-matrix. Let  $\{\lambda_i\}_{i=1}^{2n}$  be eigenvalues of the matrix  $\bar{\mathcal{H}} = \mathcal{J}\bar{\mathcal{M}}$  ordered nonincreasingly. Then it holds that*

$$\text{Re}\lambda_{2n} \leq \dots \leq \text{Re}\lambda_{n+2} < \lambda_{n+1} \leq 0 \leq \lambda_n < \text{Re}\lambda_{n-1} \leq \dots \leq \text{Re}\lambda_1.$$

Moreover, the minimal positive solution  $Y^*$  of (8) is such that  $\sigma(\bar{D} - \bar{C}Y^*) = \{\lambda_1, \dots, \lambda_n\}$  and  $\sigma(\bar{A} - Y^*\bar{C}) = \{-\lambda_{n+1}, \dots, -\lambda_{2n}\}$ .

Let  $\bar{u}$  and  $\bar{v}$  be the left and the right eigenvector of  $\bar{M}$  corresponding to the zero eigenvalue, respectively. Then  $\bar{u}$  and  $\bar{v}$  are positive and unique up to a scalar multiple. Moreover, the following statement are true: If  $\bar{u}^T \mathcal{J} \bar{v} > 0$ , then  $\lambda_n = 0$ ,  $\lambda_{n+1} < 0$ . If  $\bar{u}^T \mathcal{J} \bar{v} < 0$ , then  $\lambda_n > 0$ ,  $\lambda_{n+1} = 0$ . If  $\bar{u}^T \mathcal{J} \bar{v} = 0$ , then  $\lambda_n = \lambda_{n+1} = 0$ .

The next result, given in [9, 11], guarantees the local convergence of Newton's method applied to Ricaati equation (8).

**Theorem 3.** Let  $\bar{M}$  be a nonsingular or irreducible singular  $M$ -matrix as in Theorem 2 and  $Y^{(0)} = 0$ . Then Newton's method

$$(\bar{A} - Y^{(k)}\bar{C})Y^{(k+1)} + Y^{(k+1)}(\bar{D} - \bar{C}Y^{(k)}) = \bar{B} - Y^{(k)}\bar{C}Y^{(k)}, \quad k = 0, 1, \dots \quad (10)$$

is well defined and the matrix sequence  $\{Y^{(k)}\}$  is strictly monotonically increasing and is convergent to the minimal positive solution of (8). Furthermore, the Jacobian matrix  $\bar{M}_{Y^{(k)}} = I \otimes (\bar{A} - Y^{(k)}\bar{C}) + (\bar{D} - \bar{C}Y^{(k)})^T \otimes I$  is a nonsingular  $M$ -matrix. Particularly, if  $\bar{M}$  is nonsingular,  $\bar{A} - Y^{(k)}\bar{C}$  and  $\bar{D} - \bar{C}Y^{(k)}$  are nonsingular  $M$ -matrices.

### 3. Structured Newton's method for H-equation with $c \in (0, 1)$

We review the factored-alternating-directional-implicit (FADI) iteration method [3, 21, 27] in subsection 3.1 and describe the structured Newton's method in subsection 3.2.

#### 3.1. FADI iteration method for low-rank Sylvester equation

Let

$$PX + XQ = FG^T \quad (11)$$

be a Sylvester equation with  $P, Q \in \mathbb{R}^{n \times n}$ ,  $F, G \in \mathbb{R}^{n \times r}$  and  $r \ll n$ . Given two sets of ADI parameters  $\{p_j\}$  and  $\{q_j\}$  and some initial guess  $X^{(0)}$ . For  $j = 1, 2, \dots$  until convergence, we proceed the iteration

$$\begin{cases} (P + p_j I)X^{(j-1/2)} = X^{(j-1)}(p_j I - Q) + FG^T, \\ X^{(j)}(Q + q_j I) = (q_j I - P)X^{(j-1/2)} + FG^T. \end{cases}$$

It is not difficult to see that  $X^{(j)}$  can be expressed by  $X^{(j-1)}$  explicitly, namely,

$$\begin{aligned} X^{(j)} &= (q_j I - P)(P + p_j I)^{-1} X^{(j-1)}(p_j I - Q)(Q + q_j I)^{-1} \\ &\quad + (p_j + q_j)(P + p_j I)^{-1} FG^T (Q + q_j I)^{-1}. \end{aligned} \quad (12)$$

Let

$$T_{F_1} = \sqrt{p_1 + q_1} (P + p_1 I)^{-1} F, \quad T_{G_1} = \sqrt{p_1 + q_1} (Q + q_1 I)^{-1} G. \quad (13)$$

Then for  $j = 2, 3, \dots$ , we can rewrite (12) as the following factored form

$$\begin{cases} X^{(j)} = T_{F_j} T_{G_j}^T, \\ T_{F_j} = [\sqrt{p_j + q_j} (P + p_j I)^{-1} F, (q_j I - P)(P + p_j I)^{-1} T_{F_{j-1}}], \\ T_{G_j} = [\sqrt{p_j + q_j} (Q^T + q_j I)^{-1} G, (p_j I - Q^T)(Q^T + q_j I)^{-1} T_{G_{j-1}}]. \end{cases}$$

Since the order of the ADI parameters  $\{p_j\}$  and  $\{q_j\}$  has no significance to the ADI iteration [21], the FADI iteration method with parameters of reverse order, after  $J$  step, can be reformulated as

$$\begin{cases} X^{(J)} = T_{F_J} T_{G_J}^T, \\ T_{F_J} = [ T_{F_1}, \frac{\sqrt{p_2+q_2}}{\sqrt{p_1+q_1}} (q_1 I - P)(P + p_2 I)^{-1} T_{F_1}, \dots, \\ \quad \frac{\sqrt{p_J+q_J}}{\sqrt{p_{J-1}+q_{J-1}}} (q_{J-1} I - P)(P + p_J I)^{-1} T_{F_{J-1}} ], \\ T_{G_J} = [ T_{G_1}, \frac{\sqrt{p_2+q_2}}{\sqrt{p_1+q_1}} (p_1 I - Q^T)(Q^T + q_2 I)^{-1} T_{F_1}, \dots, \\ \quad \frac{\sqrt{p_J+q_J}}{\sqrt{p_{J-1}+q_{J-1}}} (p_{J-1} I - Q^T)(Q^T + q_J I)^{-1} T_{G_{J-1}} ], \end{cases} \quad (14)$$

where  $T_{F_1}$  and  $T_{G_1}$  are defined in (13) and  $T_{F_j} = \frac{\sqrt{p_j+q_j}}{\sqrt{p_{j-1}+q_{j-1}}} (q_{j-1} I - P)(P + p_j I)^{-1} T_{F_{j-1}}$ ,  $T_{G_j} = \frac{\sqrt{p_j+q_j}}{\sqrt{p_{j-1}+q_{j-1}}} (p_{j-1} I - Q^T)(Q^T + q_j I)^{-1} T_{G_{j-1}}$  for  $j = 2, \dots, J-1$ . We refer to [3, 21] for more details about the FADI iteration method.

### 3.2. Structured Newton's method for transformed H-equation

To highlight the implicit structure in (4), we define a Cauchy-like matrix  $X$  with

$$(X)_{ij} = \frac{x_i x_j}{\delta_i + \delta_j}.$$

Then the H-equation (4), by incorporating  $X$ , is reformulated as

$$x = \alpha \Delta X e + e \quad (15)$$

with  $\Delta = \text{diag}(\delta)$ ,  $\delta^T = (\delta_1, \dots, \delta_n)$  and  $x^T = (x_1, \dots, x_n)$ . Since the Cauchy-like matrix  $X$  satisfies the displacement equation

$$\Delta X + X \Delta = x x^T, \quad (16)$$

we substitute (15) into (16) and pre-multiply and post-multiply it by  $\Delta^{-1}$  such that (16) is equivalent to a Riccati equation

$$X C X - A X - X A^T + B = 0 \quad (17)$$

with

$$A = \Delta^{-1} - \alpha \Delta^{-1} e e^T, \quad B = \Delta^{-1} e e^T \Delta^{-1} \quad \text{and} \quad C = \alpha^2 e e^T.$$

Based on the transformed Riccati equation (17), we have the following iteration scheme.

**Structured Newton's method for H-equation with  $c \in (0, 1)$ .** Let  $\zeta = \Delta^{-1} e$  and  $\eta = \alpha e$ . Given an initial guess  $x^{(0)} = e$ , for  $k = 0, 1, 2, \dots$  until convergence proceed the iteration:

1. Let  $\xi^{(k)} = \Delta^{-1} x^{(k)}$ . Solve the following equation to get  $X^{(k+1)}$ 

$$\begin{aligned} & (\Delta^{-1} - \xi^{(k)} \eta^T) X^{(k+1)} + X^{(k+1)} (\Delta^{-1} - \eta (\xi^{(k)})^T) \\ & = (\zeta, \xi^{(k)} - \zeta) (\zeta, \zeta - \xi^{(k)})^T. \end{aligned} \quad (18)$$
2. Form the next iterative vector  $x^{(k+1)} = \alpha \Delta X^{(k+1)} e + e$ .

*Remark.* (i) Equation (18) is a Lyapunov equation and may be rewritten as (6) in a vector form, in view of the equality in step 2.

(ii) It is clear that the rank of the matrix in right hand side of (18) is 2, far less than the dimension of the equation in general, which motivates us to use the FADI iteration method to solve the subproblem (18).

(iii) The iterative matrix  $X^{(k)}$  in step 1 is unnecessary to be derived explicitly as only a vector  $x^{(k)}$  is needed at each Newton iteration.

The following result indicates that the structured Newton's method for H-equation (4) is well-defined.

**Theorem 4.** *Given the initial vector  $x^{(0)} = e$ , the sequence  $\{x^{(k)}\}_{k=1}^{+\infty}$  generated by structured Newton's method is strictly monotonically increasing and is convergent to the minimal positive solution  $x^*$  of H-equation (4).*

*Proof.* It is not difficult to see that the structured Newton's method is in essence just Newton's method applied to Riccati equation (17). Therefore by Theorem 3, we only need to show that the matrix

$$\mathcal{M} = \begin{bmatrix} A^T & -C \\ -B & A \end{bmatrix} = \begin{bmatrix} \Delta^{-1} - \alpha ee^T \Delta^{-1} & -\alpha^2 ee^T \\ -\Delta^{-1} ee^T \Delta^{-1} & \Delta^{-1} - \alpha \Delta^{-1} ee^T \end{bmatrix} \quad (19)$$

is an M-matrix. In fact, we can find two positive vectors

$$u^T = (2e^T, \frac{c}{n}e^T \Delta) \quad \text{and} \quad v^T = (ce^T \Delta, 2ne^T) \quad (20)$$

such that

$$u^T \mathcal{M} = (1 - c)(2e^T \Delta^{-1}, \frac{c}{n}e^T) > 0 \quad \text{and} \quad v^T \mathcal{M}^T = (1 - c)(ce^T, 2ne^T \Delta^{-1}) > 0$$

for  $c \in (0, 1)$ . By Lemma 1,  $\mathcal{M}$  is a nonsingular M-matrix.  $\square$

We can further show the elementwisely monotonic property in each iteration vector  $x^{(k)}$ .

**Theorem 5.** *Let the sequence  $\{x^{(k)}\}_{k=1}^{+\infty}$  be generated by structured Newton's method with an initial guess  $x^{(0)} = e$ . Then for  $k = 1, 2, \dots$ , each iteration vector  $x^{(k)}$  is elementwisely strictly monotonic, namely, for  $k = 1, 2, \dots$  the following inequalities hold true*

$$x_n^{(k)} > x_{n-1}^{(k)} > \dots > x_2^{(k)} > x_1^{(k)}. \quad (21)$$

*Proof.* Let the matrix  $S$  be defined in (5). We first show that

$$1 - (Sx^{(k)})_i > 0 \quad (22)$$

for  $i = 1, \dots, n$  and  $k = 0, 1, \dots$ . In fact, it follows from Theorem 3 that  $\Delta^{-1} - \xi^{(k)} \eta^T$  is a nonsingular M-matrix. Thus  $I - \alpha x^{(k)} e^T$  is a nonsingular M-matrix, which implies that, for any  $i = 1, \dots, n$  and  $k = 0, 1, \dots$ ,

$$1 - (Sx^{(k)})_i > 1 - \alpha e^T x^{(k)} = 1 - \rho(\alpha x^{(k)} e^T) > 0.$$

We now demonstrate the inequalities in (21) by induction on  $k$ . For  $k = 1$ , taking the  $i$ -th item of iteration vector  $x^{(k)}$  in (6) yields

$$x_i^{(1)} = \frac{1 + (S(x^{(1)} - x^{(0)}))_i}{1 - (Sx^{(0)})_i} < \frac{1 + (S(x^{(1)} - x^{(0)}))_{i+1}}{1 - (Sx^{(0)})_{i+1}} = x_{i+1}^{(1)},$$

where the inequality follows from

$$\delta_1 < \delta_2 < \dots < \delta_n,$$

(22) and Theorem 4. This proves (21) with  $k = 1$ .

Suppose that (21) is true for some  $k$ , i.e.  $x_i^{(k)} < x_{i+1}^{(k)}$  for  $i = 1, 2, \dots, n-1$ . Then by using this induction assumption and the strict monotonicity of  $\{\delta_i\}_{i=1}^n$  and  $\{x^{(k)}\}_{k=1}^{+\infty}$ , the  $i$ -th item of  $x^{(k+1)}$  in (6) satisfies

$$x_i^{(k+1)} = \frac{1 + x_i^{(k)}(S(x^{(k+1)} - x^{(k)}))_i}{1 - (Sx^{(k)})_i} < \frac{1 + x_{i+1}^{(k)}(S(x^{(k+1)} - x^{(k)}))_{i+1}}{1 - (Sx^{(k)})_{i+1}} = x_{i+1}^{(k+1)}.$$

Thus (21) holds true for all  $k$  by the principle of induction.  $\square$

As for the eigenvalues in coefficient matrices at each Newton step, we have the following interlacing property.

**Theorem 6.** *For each  $k = 1, 2, \dots$ , the coefficient matrices  $\Delta^{-1} - \xi^{(k)}\eta^T$  at the  $k$ -th Newton step have distinct eigenvalues  $\{\lambda_i^{(k)}\}_{i=1}^n$ . Moreover, they satisfy the following inequalities:*

$$0 < \lambda_1^{(k)} < \frac{1}{\delta_n} < \lambda_2^{(k)} < \frac{1}{\delta_{n-1}} < \dots < \lambda_n^{(k)} < \frac{1}{\delta_1}.$$

*Proof.* Let

$$\phi^{(k)}(z) := 1 + \sum_{i=1}^n \frac{x_i^{(k)}\alpha}{z\delta_i - 1} = 0 \quad (23)$$

be the secular equation [22] of  $\Delta^{-1} - \xi^{(k)}\eta^T - zI$  for  $k = 1, 2, \dots$ . Then  $\phi^{(k)}(x)$  is a strictly monotonically decreasing function with respect to  $z$  in the interval  $(\frac{1}{\delta_i}, \frac{1}{\delta_{i+1}})$ ,  $i = 1, \dots, n-1$ . Since for each  $i = 1, \dots, n-1$ ,  $\lim_{z \rightarrow \frac{1}{\delta_{i+1}}+} \phi^{(k)}(z) = +\infty$  and  $\lim_{z \rightarrow \frac{1}{\delta_i}-} \phi^{(k)}(z) = -\infty$ , the equation (23) has a unique real root in the open interval  $(\frac{1}{\delta_i}, \frac{1}{\delta_{i+1}})$ . Hence, we only need to show that the last root of (23) lies in  $(0, \frac{1}{\delta_n})$ . Let  $\tau$  be the minimal eigenvalue of the matrix  $\Delta^{-1} - \Delta^{-1}x^*\eta^T$ , where  $x^*$  is the minimal positive solution of H-equation (4). By an analogous proof of Lemma 2.1 in [14], we can get the fact  $0 < \tau < \frac{1}{\delta_n}$ . This together Theorem 4 yield the following inequality

$$\phi^{(k)}(\tau) = 1 + \sum_{i=1}^n \frac{x_i^{(k)}\alpha}{\tau\delta_i - 1} > 1 + \sum_{i=1}^n \frac{x_i^*\alpha}{\tau\delta_i - 1} =: \phi^*(\tau)$$



for all  $k = 1, 2, \dots$ . Since  $\tau$  is an eigenvalue of the matrix  $\Delta^{-1} - \Delta^{-1}x^*\eta^T$ , it must hold  $\phi^*(\tau) = 0$  as  $\phi^*(z) = 0$  is the secular equation of  $\Delta^{-1} - \Delta^{-1}x^*\eta^T - zI$ . So the last real root of  $\phi^{(k)}(z)$  lies in the interval  $(0, \frac{1}{\delta_n})$  due to  $\lim_{z \rightarrow \frac{1}{\delta_n}} \phi^{(k)}(z) = -\infty$ .  $\square$

*Remark.* Theorem 6 particularly shows that the coefficient matrices at each Newton step have positive extremal eigenvalues. Hence the Lyapunov equation (18) falls into the “ADI model problems”. Moreover, the diagonal-plus-rank-one structure guarantees that each eigenvalue can be evaluated in  $O(n)$  flops (see [1, 23]), so that the optimal ADI parameters could be computed readily from the two extremal eigenvalues by Wachspress's method [27, 28]. Consequently, the FADI iteration method is very efficient for solving the low rank Lyapunov equation (18).

To implement the FADI iteration method more conveniently in computation, we give the following.

**Proposition 7.** *Let  $T_F$  and  $T_G$  be factor matrices generated by the FADI iteration for solving (18). For each  $i = 1, 2, \dots, J$ , denote by  $(t_{2i-1}, t_{2i})$  and  $(\bar{t}_{2i-1}, \bar{t}_{2i})$  the  $i$ -th pair of vectors in  $T_F$  and  $T_G$ , respectively. For each  $i = 2, 3, \dots, J$ , define diagonal matrices*

$$D_{\Delta_1} = (I + p_1\Delta)^{-1}, \quad D_{\Delta_i} = (p_{i-1}\Delta - I)(I + p_i\Delta)^{-1}$$

and scalars

$$\begin{aligned} h_1 &= \frac{\eta^T D_{\Delta_1} e}{1 - \eta^T D_{\Delta_1} x^{(k)}}, & h_2 &= \frac{\eta^T D_{\Delta_1} (x^{(k)} - e)}{1 - \eta^T D_{\Delta_1} x^{(k)}}, \\ h_{2i-1} &= \frac{\eta^T (I + p_i\Delta)^{-1} \Delta t_{2i-3}}{1 - \eta^T (I + p_i\Delta)^{-1} x^{(k)}}, & h_{2i} &= \frac{\eta^T (I + p_i\Delta)^{-1} \Delta t_{2i-2}}{1 - \eta^T (I + p_i\Delta)^{-1} x^{(k)}} \end{aligned}$$

with  $\eta = \alpha e$ . Then we have

$$\begin{bmatrix} t_1, & t_2 \end{bmatrix} = \sqrt{2p_1} \begin{bmatrix} D_{\Delta_1}(e + h_1 x^{(k)}), & D_{\Delta_1}((1 + h_2)x^{(k)} - e) \end{bmatrix}, \quad (24)$$

$$\begin{bmatrix} \bar{t}_1, & \bar{t}_2 \end{bmatrix} = \begin{bmatrix} t_1, & -t_2 \end{bmatrix} \quad (25)$$

and for  $i = 2, \dots, J$

$$\begin{aligned} \begin{bmatrix} t_{2i-1}, & t_{2i} \end{bmatrix} &= \sqrt{2p_i} \begin{bmatrix} D_{\Delta_i}(t_{2i-3} + h_{2i-1}\xi^{(k)}) + h_{2i-1}\xi^{(k)}, \\ D_{\Delta_i}(t_{2i-2} + h_{2i}\xi^{(k)}) + h_{2i}\xi^{(k)} \end{bmatrix}, \end{aligned} \quad (26)$$

$$\begin{bmatrix} \bar{t}_{2i-1}, & \bar{t}_{2i} \end{bmatrix} = \begin{bmatrix} t_{2i-1}, & -t_{2i} \end{bmatrix} \quad (27)$$

with  $\xi^{(k)} = \Delta^{(-1)}x^{(k)}$ .

*Proof.* For  $i = 1, \dots, J$ , by using the Sherman-Morrison-Woodbury formula (see [8] for example) directly to the matrices

$$(P + p_i I)^{-1} = (\Delta^{-1} + p_i I - \xi^{(k)}\eta^T)^{-1}$$

in (14), we have equalities (24) and (26). Note that  $Q^T = P$  in (14) for Riccati equation (17), equalities (25) and (27) hold true.  $\square$

Notice that, when solving the subproblem (18), the iteration matrix  $X^{(k)}$  must not to be formed exactly since only the a matrix-vector product (i.e.  $X^{(k)}e$ ) is needed. So the FADI iteration method, as mentioned before, can be implemented in a low storage way. Specifically, in current FADI iteration step, we use two vectors to store  $[t_{2i-1}, t_{2i}]$  and a temporary vector  $x_{temp}$  to store the sum of first  $2i$  items in  $x^{(k)}$ , then update the two storage vectors with  $[t_{2i+1}, t_{2i+2}]$  and add them to  $x_{temp}$  in next FADI iteration step. When the  $J_k$ -th FADI iteration step is completed,  $x_{temp}$  approximates the exact  $x^{(k)}$  in the  $k$ -th Newton step. This approach avoids constructing the two  $n \times 2J$  factor matrices  $T_F$  and  $T_G$  in (14). We give all steps of the structured Newton-FADI iteration method for H-equation with  $c \in (0, 1)$  in Algorithm 1.

---

---

**Algorithm 1.**

Inputs: Initial guess  $x^{(0)} = e$  and tolerance  $tol$ .

Outputs:  $x \in \mathbb{R}^n$  is approximative minimal positive solution of H-equation.

---

1.  $x_{temp} := x^{(0)}$ .
  2. For  $k = 0, 1, 2, \dots$ , until convergence, do
  3.   Compute the two extremal eigenvalues of  $\Delta^{-1} - \Delta^{-1}(x_{temp})\eta^T$ .
  4.   Determine the number of ADI iterations  $J_k$  and compute the  
           optimal ADI parameters  $\{p_j\}_{j=1}^{J_k}$ .
  5.   Form a pair of vectors  $[t_1, t_2]$  using (24).
  6.    $x_{temp} := (t_1^T e)t_1 - (t_2^T e)t_2$ .
  7.   For  $i = 2, 3, \dots, J_k$ , do
  8.       Update vectors  $[t_1, t_2]$  with formula (26).
  9.        $x_{temp} := x_{temp} + (t_1^T e)t_1 - (t_2^T e)t_2$ .
  10.   End
  11.    $x_{temp} := \alpha \Delta x_{temp} + e$ .
  12.   Evaluate  $\|F(x_{temp})\|_2$ . If  $\|F(x_{temp})\|_2 < tol$ ,  $x := x_{temp}$ , stop.
  13.    $k := k + 1$ .
  14.   End
- 
- 

*Remark.* (i) We compute the two extremal eigenvalues in step 3 by a similar method in [23]. In addition, we adopted the so-called “warm start” strategy in the computation of the extremal eigenvalues of the successive coefficient matrices. More precisely, except the first iteration, we chose the currently obtained minimal (maximal) eigenvalue as the initial guess in the iterative process for the next minimal (maximal) eigenvalue. (ii) The method to determine  $J_k$  and select the optimal ADI parameters  $\{p_j\}$  in step 4 can be found in [21, 27, 28]. (iii) During iterations in Algorithm 1, we evaluate the H-equation in step 12 only once to test that the stop criterion is satisfied or not. The choice of the tolerance for the termination is given in Section 5.

#### 4. The shift technique for H-equation with $c = 1$

In this section, we consider the structured Newton’s method for H-equation

with the parameter  $c = 1$ . In this case, the structured Newton's method for H-equation given in last section may not retain the quadratic convergence since the Jacobian at the solution is singular. In fact, if we still let the matrix  $\mathcal{M}$  and vectors  $u, v$  be defined in (19) and (20), respectively, it is clear that  $u$  and  $v$  are the left and the right eigenvector of  $\mathcal{M}$  corresponding to the zero eigenvalue when  $c = 1$ . Moreover it follows from Theorem 2 that the matrix  $\mathcal{H} := \mathcal{J}\mathcal{M}$  has two zero eigenvalues since  $u^T \mathcal{J}v = 0$ . That is, both spectrums of  $A - X^*C$  and  $A^T - CX^*$  contain a zero eigenvalue, respectively, where  $X^*$  is the minimal positive solution of Riccati equation (17). Consequently, the sequence of Jacobian  $\{\mathcal{M}_{X^{(k)}}\}$  in Newton's method converges to a singular M-matrix  $\mathcal{M}_{X^*} = I \otimes (A - X^*C) + (A^T - CX^*)^T \otimes I$ .

The shift technique, introduced in [12] (see also [2, 10]), aims at removing the singularity of the matrix  $\mathcal{M}_{X^*}$  by converting a zero eigenvalue to a nonzero one so that Newton's method may recover the quadratic convergence. We will exploit this technique to cope with the Riccati equation (17) with  $c = 1$ . To avoid notational clutter, we still use the same symbols as in Theorem 2.

Let  $\bar{\mathcal{H}} = \mathcal{H} + vv^T$  with  $w^T := (w_1^T, w_2^T) = (\frac{1}{2n}e^T \Delta^{-1}, \frac{1}{4n^2}e^T)$  and  $v^T := (v_1^T, v_2^T) = (e^T \Delta, 2ne^T)$ . Then we have  $w^T v = 1$ . By Lemma 5.3 in [10], the eigenvalues of  $\bar{\mathcal{H}}$  are those of  $\mathcal{H}$  except one zero eigenvalue replaced by 1. Denote the subblocks of  $\bar{\mathcal{H}}$  by

$$\bar{A} = A - v_2 w_2^T, \quad \bar{B} = B + v_2 w_1^T, \quad \bar{C} = C - v_1 w_2^T, \quad \bar{D} = A^T + v_1 w_1^T.$$

Then we can construct a new Riccati equation

$$Y\bar{C}Y - \bar{A}Y - Y\bar{D} + \bar{B} = 0. \quad (28)$$

It follows from Theorem 5.4 in [10] that the Riccati equation (28) shares the same minimal positive solution (i.e.  $X^*$ ) with the Riccati equation (17) and  $\sigma(\bar{D} - \bar{C}X^*) = \{\lambda_1, \dots, \lambda_{n-1}, 1\}$ . Therefore, we can implement the Newton's method for finding the minimal positive solution of new Riccati equation (28) instead of the original one (17). This leads the following iterative scheme for H-equation with  $c = 1$ .

**Structured Newton's method for H-equation with  $c = 1$ .** Let  $\eta = \alpha e$ ,  $\gamma = \alpha(I - \Delta)e$ ,  $\beta = (\Delta^{-1} + I)e$  and  $\zeta = \Delta^{-1}e$ . Given an initial guess  $Y^{(0)} = 0$ , for  $k = 0, 1, 2, \dots$  until convergence proceed the iteration:

1. Let  $a^{(k)} = \alpha Y^{(k)}(I - \Delta)e$ ,  $b^{(k)} = \alpha(Y^{(k)})^T e$ . Solve the equation
$$(\Delta^{-1} - (\beta + a^{(k)})\eta^T)Y^{(k+1)} + Y^{(k+1)}(\Delta^{-1} - \gamma(\zeta + b^{(k)})^T) = (\beta, -a^{(k)})(\zeta, b^{(k)})^T$$
to get  $Y^{(k+1)}$ . (29)
2. Form the next iteration vector  $y^{(k+1)} = \alpha \Delta Y^{(k+1)}e + e$ .

Similar to the proof of Theorem 4, we can show that the above structured Newton's method is well-defined. In fact, if we let  $\bar{\mathcal{M}} = \mathcal{J}\bar{\mathcal{H}}$ , then it is a Z-matrix and

$$u^T \bar{\mathcal{M}} = u^T \mathcal{J}\bar{\mathcal{H}} = u^T \mathcal{M} + u^T \mathcal{J}vv^T = 0,$$

where  $u$  is a positive vector defined in (20). As  $\bar{\mathcal{M}}$  is irreducible, it follows from Theorem 3 that the above structured Newton's method is well defined.

By applying FADI iteration method into the Sylvester equation (29), we have the following concise formula in (14). We omit the proof as it is analogous to that in Proposition 7.

**Proposition 8.** *Let  $T_F$  and  $T_G$  be factor matrices generated by the FADI iteration for solving (29). For each  $i = 1, 2, \dots, J$ , denote by  $(t_{2i-1}, t_{2i})$  and  $(\bar{t}_{2i-1}, \bar{t}_{2i})$  the  $i$ -th pair of vectors in  $T_F$  and  $T_G$ , respectively. For each  $i = 2, 3, \dots, J$ , define diagonal matrices*

$$\begin{aligned} D_{\Delta_1} &= (\Delta^{-1} + p_1 I)^{-1}, & \bar{D}_{\Delta_1} &= (\Delta^{-1} + q_1 I)^{-1}, \\ D_{\Delta_i} &= (q_{i-1} I - \Delta^{-1})(\Delta^{-1} + p_i I)^{-1}, & \bar{D}_{\Delta_i} &= (p_{i-1} I - \Delta^{-1})(\Delta^{-1} + q_i I)^{-1} \end{aligned}$$

and scalars

$$\begin{aligned} h_1 &= \frac{\eta^T D_{\Delta_1} \beta}{1 - \eta^T D_{\Delta_1} (\beta + a^{(k)})}, & h_2 &= \frac{\eta^T D_{\Delta_1} a^{(k)}}{1 - \eta^T D_{\Delta_1} (\beta + a^{(k)})}, \\ \bar{h}_1 &= \frac{\gamma^T \bar{D}_{\Delta_1} \zeta}{1 - \gamma^T \bar{D}_{\Delta_1} (\zeta + b^{(k)})}, & \bar{h}_2 &= \frac{\gamma^T \bar{D}_{\Delta_1} b^{(k)}}{1 - \gamma^T \bar{D}_{\Delta_1} (\zeta + b^{(k)})}, \\ h_{2i-1} &= \frac{\eta^T (p_i I + \Delta^{-1})^{-1} t_{2i-3}}{1 - \eta^T (p_i I + \Delta^{-1})^{-1} (a^{(k)} + \beta)}, & h_{2i} &= \frac{\eta^T (p_i I + \Delta^{-1})^{-1} t_{2i-2}}{1 - \eta^T (p_i I + \Delta^{-1})^{-1} (a^{(k)} + \beta)}, \\ \bar{h}_{2i-1} &= \frac{\gamma^T (q_i I + \Delta^{-1})^{-1} \bar{t}_{2i-3}}{1 - \gamma^T (q_i I + \Delta^{-1})^{-1} (b^{(k)} + \zeta)}, & \bar{h}_{2i} &= \frac{\gamma^T (q_i I + \Delta^{-1})^{-1} \bar{t}_{2i-2}}{1 - \gamma^T (q_i I + \Delta^{-1})^{-1} (b^{(k)} + \zeta)}. \end{aligned}$$

Then we have

$$[t_1, t_2] = \sqrt{p_1 + q_1} [D_{\Delta_1} ((1 + h_1)\beta + h_1 a^{(k)}), D_{\Delta_1} ((h_2 - 1)a^{(k)} + h_2 \beta)], \quad (30)$$

$$[\bar{t}_1, \bar{t}_2] = \sqrt{p_1 + q_1} [\bar{D}_{\Delta_1} (1 + \bar{h}_1)\zeta + \bar{h}_1 b^{(k)}, \bar{D}_{\Delta_1} ((1 + \bar{h}_2)b^{(k)} - \bar{h}_2 \zeta)] \quad (31)$$

and for  $i = 2, \dots, J$

$$\begin{aligned} [t_{2i-1}, t_{2i}] &= \sqrt{p_i + q_i} [D_{\Delta_i} (t_{2i-3} + h_{2i-1}(\beta + a^{(k)})) + h_{2i-1}(\beta + a^{(k)}), \\ &\quad D_{\Delta_i} (t_{2i-2} + h_{2i}(a^{(k)} + \beta)) + h_{2i}(a^{(k)} + \beta)], \quad (32) \end{aligned}$$

$$\begin{aligned} [\bar{t}_{2i-1}, \bar{t}_{2i}] &= \sqrt{p_i + q_i} [\bar{D}_{\Delta_i} (\bar{t}_{2i-3} + \bar{h}_{2i-1}(\zeta + b^{(k)})) + \bar{h}_{2i-1}(\zeta + b^{(k)}), \\ &\quad \bar{D}_{\Delta_i} (\bar{t}_{2i-2} + \bar{h}_{2i}(\zeta + b^{(k)})) + \bar{h}_{2i}(\zeta + b^{(k)})]. \quad (33) \end{aligned}$$

We list all steps of the structured Newton's method for H-equation with  $c = 1$  in Algorithm 2, noting that only two pairs of vectors are used for the storage and

the update in the FADI iteration.

---



---

**Algorithm 2.**

Inputs: Initial guess  $y^{(0)} = 0$  and tolerance  $tol$ .

Outputs:  $y \in \mathbb{R}^n$  is an approximative positive solution of H-equation.

---

1.  $a = b = 0$ ,  $\gamma = \alpha(I - \Delta)e$ ,  $\eta = \alpha e$ ,  $\zeta = \Delta^{-1}e$ ,  $\beta = (\Delta^{-1} + I)e$ .
  2. For  $k = 0, 1, 2, \dots$ , until convergence, do
  3. Find extremal eigenvalues of  $\Delta^{-1} - (\beta - a)\eta^T$ ,  $\Delta^{-1} - \gamma(\zeta + b)^T$ .
  4. Compute the number of ADI iterations  $J_k$  and the optimal ADI parameters  $\{p_j\}_{j=1}^{J_k}$  and  $\{q_j\}_{j=1}^{J_k}$ .
  5. Form vectors  $[t_1, t_2]$  and  $[\bar{t}_1, \bar{t}_2]$  using (30) and (31).
  6.  $a := (\bar{t}_1^T \gamma)t_1 + (\bar{t}_2^T \gamma)t_2$ .
  7.  $b := (t_1^T \eta)\bar{t}_1 + (t_2^T \eta)\bar{t}_2$ .
  8.  $y_{temp} := (\bar{t}_1^T \eta)t_1 + (\bar{t}_2^T \eta)t_2$ .
  9. For  $i = 2, 3, \dots, J_k$ , do
  10. Update vectors  $[t_1, t_2]$  and  $[\bar{t}_1, \bar{t}_2]$  with (32) and (33).
  11.  $a := a + (\bar{t}_1^T \gamma)t_1 + (\bar{t}_2^T \gamma)t_2$ .
  12.  $b := b + (t_1^T \eta)\bar{t}_1 + (t_2^T \eta)\bar{t}_2$ .
  13.  $y_{temp} := y_{temp} + (\bar{t}_1^T \eta)t_1 + (\bar{t}_2^T \eta)t_2$ .
  14. End
  15.  $y_{temp} := \Delta y_{temp} + e$ .
  16. Evaluate  $\|F(y_{temp})\|_2$ . If  $\|F(y_{temp})\|_2 < tol$ ,  $y := y_{temp}$ , stop.
  17.  $k := k + 1$ .
  18. End
- 
- 

## 5. Numerical experiments

The purpose of this section is to show the effectiveness of Algorithm 1 and 2 in computation. The test problems come from the discrete integral equation (3) with different  $c \in (0, 1]$ , see [18] for more details.

Our algorithms are coded by MATLAB 7.1 and are run on a PC with 1.80 GHz AMD Sempron 3000+ processor and 512M RAM.

We compare the performance of Algorithms 1 and 2 with that of Newton-Krylov methods (Newton-GMRE (NGMRE), Newton-BiCGSTAB (NBICG) and Newton-TFQMR (NTRQMR) in [18]), the MATLAB codes of which can be downloaded from Kelley's web page at

<http://www4.ncsu.edu/~ctk/newtony.html>.

The termination criterion for all algorithms is to stop the iteration if the inequality

$$\|F(x^{(k)})\| \leq \tau_r \|F(x^{(0)})\| + \tau_a$$

is satisfied, where  $\tau_r$  and  $\tau_a$  are the relative error tolerance and absolute error tolerance [18]. In our experiments, we take  $\tau_r = \tau_a = 10^{-12}$ .

**Table 1 Test Results for  $c = 0.5$  and  $c = 0.9$** 

$n$	Method	$c = 0.5$				$c = 0.9$			
		ALG.1	NGMRE	NBICG	NTRQME	ALG.1	NGMRE	NBICG	NTRQME
1000	CPU	0.07	0.14	0.09	0.07	0.04	0.10	0.12	0.10
	IT	4	6	5	6	5	6	6	6
	FE	4	14	16	21	5	17	23	27
	NMF	7.83e-15	2.22e-12	5.98e-14	5.73e-14	2.84e-14	6.19e-13	4.15e-14	5.01e-14
2000	CPU	0.15	0.28	0.31	0.28	0.18	0.32	0.45	0.42
	IT	4	6	5	6	5	6	6	6
	FE	4	14	16	21	5	17	23	27
	NMF	1.44e-14	3.14e-12	8.53e-14	8.19e-14	6.64e-14	8.73e-13	7.81e-14	9.12e-14
3000	CPU	0.31	0.60	0.67	0.64	0.37	0.73	1.00	0.92
	IT	4	6	5	6	5	6	6	6
	FE	4	14	16	21	5	17	23	27
	NMF	2.12e-14	3.84e-12	1.05e-13	1.02e-13	8.32e-14	1.06e-12	1.20e-13	1.24e-13
4000	CPU	0.50	1.07	1.21	1.14	0.64	1.34	1.71	1.67
	IT	4	6	5	6	5	6	6	6
	FE	4	14	16	21	5	17	23	27
	NMF	2.84e-14	4.44e-12	1.24e-13	1.19e-13	1.16e-13	1.22e-12	1.56e-13	1.68e-13

**Table 2 Test Results for  $c = 0.9999$  and  $c = 0.999999$** 

$n$	Method	$c = 0.9999$				$c = 0.999999$			
		ALG.1	NGMRE	NBICG	NTRQME	ALG.1	NGMRE	NBICG	NTRQME
1000	CPU	0.10	0.15	0.23	0.25	0.15	0.21	0.32	0.34
	IT	10	9	9	9	13	12	12	12
	FE	10	30	46	54	13	42	61	75
	NMF	1.11e-13	3.98e-12	2.16e-13	6.09e-12	7.71e-12	6.39e-12	6.17e-13	5.63e-12
2000	CPU	0.35	0.57	0.89	0.87	0.46	0.82	1.17	1.28
	IT	10	9	9	9	13	12	12	12
	FE	10	30	46	54	13	42	61	75
	NMF	2.14e-13	5.62e-12	3.35e-13	8.60e-12	1.11e-11	9.12e-12	9.05e-13	8.03e-12
3000	CPU	0.75	1.29	1.95	1.95	0.96	1.79	2.59	2.82
	IT	10	9	9	9	13	12	12	12
	FE	10	30	46	54	13	42	61	75
	NMF	3.17e-13	6.92e-12	4.26e-13	1.05e-11	1.35e-11	1.15e-11	1.14e-12	9.83e-12
4000	CPU	1.18	2.28	3.48	3.46	1.56	3.23	4.64	5.00
	IT	10	9	9	9	13	12	12	12
	FE	10	30	46	54	13	42	61	75
	NMF	5.36e-13	7.95e-12	5.34e-13	1.22e-11	1.56e-11	1.28e-11	1.33e-12	1.13e-11

We first test Algorithm 1, NGMRE, NBICG and TFQMR for the problem with  $c = 0.5, 0.9, 0.9999$  and  $0.999999$ . The test results are listed in Tables 1–2, 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 number of iterations in Newton’s method, the “FE” row records the number of evaluations of H-equation and, the “NMF” row in tables is the 2-norm of the H-equation at the approximative minimal positive solution obtained by each algorithm.

We see clearly from results in Tables 1 that when  $c$  is not (or not very) close to 1, the performance Algorithms 1 is better than the other three algorithms in CPU time, function evaluations and accuracy. The results in Table 2 show that as the parameter  $c$  approaches to 1, Algorithm 1, compared with other algorithms, needs more iterations but less CPU time and function evaluations to attain the prescribed accuracy.

We then compare the performance of Algorithms 2 and other three algorithms when  $c = 1$ . The results are listed in Table 3. Obviously, we can see from the table that Algorithm 2 performs much better than other three algorithms in CPU time,

**Table 3** Test Results for  $c = 1$ 

$n$	Method	$c = 1$			
		ALG.2	NGMRE	NBICG	NTRQME
1000	CPU	0.12	0.48	0.64	0.73
	IT	5	21	21	21
	FE	5	84	122	156
	NMF	7.79e-14	4.28e-12	2.81e-12	3.98e-12
2000	CPU	0.32	1.65	2.32	2.70
	IT	5	21	21	21
	FE	5	84	122	156
	NMF	1.58e-13	6.10e-12	4.03e-12	5.65e-12
3000	CPU	0.60	3.62	5.26	6.17
	IT	5	21	21	21
	FE	5	84	122	156
	NMF	2.24e-13	7.48e-12	4.95e-12	6.99e-12
4000	CPU	0.92	6.45	9.25	10.70
	IT	5	21	21	21
	FE	5	84	122	156
	NMF	3.15e-13	8.75e-12	5.72e-12	8.16e-12

iterative number and accuracy. This indicates Algorithm 2 is very efficient to obtain the minimal positive solution of H-equation in the critical case.

## 6. Conclusions

We have proposed a structured Newton's method for solving Chandrasekhar H-equation. The method exploits the implicit structure sufficiently to decrease the complexity of original Newton's method. The preliminary numerical results show that the proposed method outperforms some well-know Newton-Krylov methods especially in the critical case. A possible future topic is the generalization of the new-presented method to more complicated H-equations such as in [13].

**Acknowledgement.** This work was supported in part by the major project of the Ministry of Education of China granted 309023.

## R E F E R E N C E S

- [1] *D. A. Bini, L. Gemignani and V. Y. Pan*, Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equations, *Numer. Math.*, **100** (2005), 373–408.
- [2] *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), 276–290.
- [3] *P. Benner, R. C. Li and N. Truhar*, On the ADI method for Sylvester equations, *J. Comp. Appl. Math.*, **233** (2009), 1035–1045.
- [4] *S. Chandrasekhar*, Radiative transfer, Dover, New York, 1960.
- [5] *E. Y. Deeba and S. A. Khuri*, The decomposition method applied to Chandrasekhar H-equation, *Appl. Math. Comput.*, **1** (1996), 67–78.
- [6] *K. I. Dickson, C. T. Kelley, I. C. F. Ipsen and I. G. Kevrekidis*, Condition Estimates for Pseudo-arclength continuation, *SIAM J. Numer. Anal.*, **45** (2008), 263–276.
- [7] *R. W. Freund*, A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Comput.*, **14** (1993), 470–482.

- 
- [8] *G. H. Golub and C. F. Van Loan*, Matrix computations, (3rd Edition), Johns Hopkins University Press, Baltimore, MD, (1996).
  - [9] *C.-H. Guo*, Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices, SIAM J. Matrix Anal. Appl., **23** (2001), 225–242.
  - [10] *C.-H. Guo, B. Iannazzo and B. Meini*, On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equations, SIAM J. Matrix Anal. Appl., **29** (2007), 1083–1100.
  - [11] *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), 376–391.
  - [12] *C. He, B. Meini and N. H. Rhee*, A shifted cyclic reduction algorithm for quasi-birth-death problems, SIAM J. Matrix Anal. Appl. **23** (2001), 673–691.
  - [13] *J. Juang*, On coupled integral H-like equations of Chandrasekhar, SIAM J. Math. Anal. **4** (1995), 869–879.
  - [14] *J. Juang and W.-W. Lin*, Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices, SIAM J. Matrix Anal. Appl. **20** (1998), 228–243.
  - [15] *T. Kailath, A. H. Sayed*, Fast reliable algorithms for matrices with structure, Society for Industrial and Applied Mathematics, Philadelphia, (1999).
  - [16] *C. T. Kelley*, A comparison of iteration schemes for Chandrasekhar H-equations in multigroup neutron transport, J. Math. Phys., **21** (1980), 408–409.
  - [17] *C. T. Kelley*, Solution of the Chandrasekhar H-equation by Newton’s method, J. Math. Phys., **21** (1980), 1625–1628.
  - [18] *C. T. Kelley*, Solving nonlinear equations with Newton’s method, Society for Industrial and Applied Mathematics, Philadelphia, (2003).
  - [19] *C. T. Kelley, I. G. Kevrekidis, and L. Qiao*, Newton-Krylov solvers for time-steppers, Tech. rep. CRSC-TR04-10, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, (2004).
  - [20] *S. A. Khuri*, On the solution of coupled H-like equations of Chandrasekhar, Appl. Math. Comput., **2-3** (2002), 479–485.
  - [21] *J.-R. Li and J. White*, Low-rank solution of Lyapunov equations, SIAM J. Matrix Anal. Appl., **24** (2002), 260–280.
  - [22] *R.-C. Li*, Solving secular equations stably and efficiently, Technical Report, Department of Mathematics, University of California, Berkeley, CA, LAPACK Working Note **89**, (1993).
  - [23] *V. Mehrmann and H.-G. Xu*, Explicit solutions for a Riccati equation from transport theory, SIAM J. Matrix Anal. Appl. **30** (2008), 1339–1357.
  - [24] *Y. Saad and M. Schultz*, GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Comput., **7** (1986), 856–869.
  - [25] *R. Varga*, Matrix iterative analysis, (2nd Edition), Springer-Verlag Berlin Heidelberg, (2000).
  - [26] *H. A. Van der Vorst*, Bi-CGSTAB: A fast and smoothly converging variant to Bi-CG for the solution of nonsymmetric systems, SIAM J. Sci. Comput., **13** (1992), 631–644.
  - [27] *E. L. Wachspress*, The ADI model problem, Windsor, CA, (1995).
  - [28] *E. L. Wachspress*, ADI iteration parameters for solving Lyapunov and Sylvester equations, Technical Report, March, (2009).