Proceedings of the Estonian Academy of Sciences,
2020, **69**, 2, 97–108
https://doi.org/10.3176/proc.2020.2.01
Available online at www.eap.ee/proceedings

COMPUTATIONAL MATHEMATICS

Structured Shamanskii methods for Chandrasekhar equation arising from radiation

Ning Dong, Bo Yu*, and Zhaoyun Meng

School of Science, Hunan University of Technology, Zhuzhou, 412008, China

Received 27 September 2019, accepted 14 January 2020, available online 16 March 2020

Abstract. The Chandrasekhar equation describes the particles emerging from the atmospheric radiation and its solution of physical significance is the minimal positive solution. This paper analyses the efficiency index of Newton's iteration in detail, which then helps to design a structured Shamanskii method for calculating the minimal positive solution. The monotone convergence of the presented algorithm is subsequently established as well as the elementary monotonicity of the solution. Preliminary numerical experiments are listed to indicate that the newly developed two-step Shamanskii method outperforms the Newton's method in terms of CPU time and iterative number with almost no loss in accuracy.

Key words: Chandrasekhar equation, Newton's method, structured Shamanskii method, factor-alternating direction implicit iteration.

1. INTRODUCTION

The Chandrasekhar's H-function

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

with $c \in [0,1]$ (the average total number of particles emerging from a collision), was first introduced by Subrahmanyan Chandrasekhar [3,4,13]. The solution associated with problems involving scattering in atmospheric radiation was also proposed by him.

To calculate the Chandrasekhar's H-function in conservative case, the equation

$$F(H)(\delta) = 0 \tag{1}$$

has to be solved by numerical methods. Exploiting the numerical integration formula using the Composite Midpoint Rule

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

* Corresponding author, boyu_hut@126.com

with $\delta_i = (i-1/2)/n$ for $1 \leq i \leq n$ yields the nonlinear vector equation $F(x) = 0$ with its i-th element

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

The practical solution in physics is the minimal positive solution which has been studied in [11,13]. Various iterative methods such as fixed-point iterations and Newton's methods [5,13,15] were employed to find the minimal positive solution. Especially in case of Newton's method, the subproblem could be solved efficiently by some Krylov subspace methods [14]. An interesting property which has not been fully explored regarding these methods is the solution of Eq. (1), which closely relates to the Cauchy-like matrix

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

that implies the structured numerical low-rank factorization. In fact, the Chandrasekhar equation (1) with the structure presented above can be rewritten as a matrix Riccati equation

$$\mathscr{R}(X) = XCX - AX - XA^T + B = 0, \tag{3}$$

where the matrix

$$K = \begin{pmatrix} A^T & -C \\ -B & A \end{pmatrix} \tag{4}$$

is a nonsingular M-matrix or an irreducible singular M-matrix with $A = \Delta^{-1}(I - \alpha ee^T)$, $B = \Delta^{-1}ee^T\Delta^{-1}$, $C = \alpha^2 ee^T$, $\Delta = \mathrm{diag}(\delta_1, ..., \delta_n)$; and $e = (1,...,1)^T$.

For given initial value $X^{(0)}$ with the ij element $\frac{x_i^{(0)}x_j^{(0)}}{\delta_i+\delta_j} = 0$, Newton's method for Riccati equation (3) admits the following iteration format [1,8–10,12,18]

$$(A - X^{(k)}C)X^{(k+1)} + X^{(k+1)}(A - X^{(k)}C)^T = B - X^{(k)}CX^{(k)}, \;\; k \geq 0. \tag{5}$$

Note that, by introducing $x^{(k)} = \alpha\Delta X^{(k)}e + e$, the matrix $B - X^{(k)}CX^{(k)}$ on the right hand side (RHS) is a product of two rank-two matrices $(\Delta^{-1}(e, x^{(k)} - e))(\Delta^{-1}(e, e - x^{(k)}))^T$; and $A - X^{(k)}C$ on the left hand side (LHS) is a diagonal-plus-rank-one matrix $\Delta^{-1} - \alpha\Delta^{-1}x^{(k)}e^T$, then each $X^{(k+1)}$ in the above structured Newton step might be expressed implicitly and stored in low-rank form via the factor-alternating direction implicit (FADI) method [2,16]. The obvious advantage of using FADI method to solve subproblems, compared to those equipped with Krylov subspace methods [14], is that the complexity of solving the subproblem could be down to $O(n)$ [16,20,24]. However, from the viewpoint of computational efficiency, the above structured Newton's method might still leave enough room for further improvement. To see concretely, one may employ an evaluation of efficiency index (EI)

$$\mathrm{EI} = p^{1/q}$$

of an iterative method, where p is the order of the method and q represents the number of pieces of information for one iteration [19]. For example, the classic Newton's method according to the EI above, has the efficiency $\sqrt{2} = 1.4142$, since it converges quadratically with the requirement of two pieces of information (one function evaluation and one derivative evaluation). Regarding iterations in the form of vector or matrix, an evaluation index in terms of flop counts is more appropriate [6,9]. For structured Newton's method, the computation cost for each iteration $x^{(k)}$ is about $(21J_k - 10)n$, where J_k is the number of FADI iterations in each Newton step. Considering $(10.5J_k - 5)n$ as a piece of information, the EI of structured Newton's method is still 1.4142. However, enhanced convergence of Newton's method with the available Fréchet derivative $F'(x^{(k)})$ may potentially result in improved efficiency. Actually, the complexity of implementing

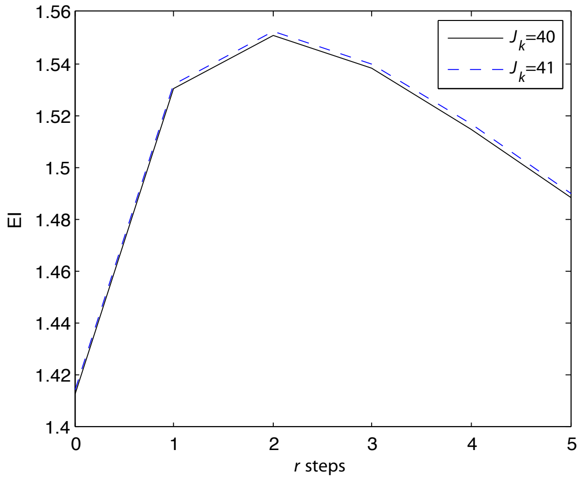

Fig. 1. Function of the efficiency index (EI).

chord method for r times in current step takes $(6J_k-1)n$ and the number J_k in all structured Newton steps varies generally between 40 and 41. So, the efficiency index of r-step Shamanskii method is

$$\mathrm{EI}=(2+r)^{\frac{10.5J_k-5}{21J_k-10+(6J_k-1)r}}.$$

Obviously, EI is 1.4142 when $r=0$ and it is just the original structured Newton's method. One interesting question is – when will the value of EI with increasing r attain its peak? To see this, two different cases ($J_k=40$ and $J_k=41$) of EI vs r were plotted in Fig. 1.

It is clear from Fig. 1 that in both cases, $J_k=40$ or $J_k=41$, EI has the maximal efficiency value around 1.55 when r = 2. This contributes to the motivation of designing the structured Shamanskii method for solving Eq. (1). Numerical experiments show that the modified Shamanskii method with $r=2$ actually outperforms the structured Newton's method in terms of CPU time and iteration number without compensating any residual accuracy.

Throughout this paper we use "$\circ$" to denote Hadamard product and "$(A)_{ij}$" or "$(v)_i$" to stand for the element of a matrix $A\in\mathbb{R}^{n\times n}$ or a vector $v\in\mathbb{R}^n$. Let I be the identity matrix of order 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 non-singular M-matrix if $s=\rho(B)$ or $s>\rho(B)$, where $\rho(\cdot)$ is the spectral radius.

The following several lemmas about some properties of M-matrices and the Riccati equation (3) are also required in this paper.

Lemma 1. ([21]). *For a Z-matrix A, it is true that*
(i) *A is an M-matrix if and only if $u^TA\geq 0$ ($Av\geq 0$) for some vector $u\geq 0$ ($v\geq 0$);*
(ii) *when A is non-singular, A is an M-matrix if and only if $A^{-1}\geq 0$ or, $u^TA>0$ ($Av>0$) for some vector $u>0$ ($v>0$) or, the spectrum $\sigma(A)\in\mathbb{C}^+$ with $\mathbb{C}^+$ denoting the open right half-plane.*

Lemma 2. ([8,9]). *The Riccati equation* (3) *has a minimal nonnegative solution X^*. If K is irreducible, then $X^*>0$ and $A-X^*C$ is an irreducible M-matrix. If K is a nonsingular M-matrix, then $A-X^*C$ and is a nonsingular M-matrix. If K is an irreducible singular M-matrix, there exist positive vectors $u_1,v_1\in\mathbb{R}^n$ and $u_2,v_2\in\mathbb{R}^n$ such that*

$$K(v_1^T,v_2^T)^T=0,\ \ (u_1^T,u_2^T)K=0,$$

and the vectors (v_1^T,v_2^T) and (u_1^T,u_2^T) are unique up to a scalar multiple, the matrix $A-X^*C$ in this case is a singular M-matrix.

The rest of this paper is organized so that the review of the Shamanskii method and its structured version is presented in Section 2. Section 3 focuses on the establishment of the monotone convergence and the elementary monotonicity of the solution. We implement the numerical experiments to show the effectiveness of the structured Shamanskii method in Section 4 and draw the conclusions in Section 5.

2. STRUCTURED SHAMANSKII METHOD

There are different versions of Newton's iterations with higher order convergence, of which the Shamanskii method is one of the most important. Its essence is to impose several chord iterations in each Newton step so that the precipitation is possible. The optimal number of chord iterations in Shamanskii method might be 2 due to the EI interpretation described in introduction. However, r-step structured Shamanskii method will be described here for more general purpose.

2.1. Structured Shamanskii method

1. Given initial vector $x^{(0)} = 0$ and $(X^{(0)})_{ij} = \frac{x_i^{(0)} x_j^{(0)}}{\delta_i+\delta_j}$, for $k = 0,1,2,...$, set $Y_0^{(k)} = X^{(k)}$ and solve the equation for $m = 0,...r-1$:

$$(A - X^{(k)}C)Y_{m+1}^{(k)} + Y_{m+1}^{(k)}(A - X^{(k)}C)^T = B - X^{(k)}CX^{(k)} + (Y_m^{(k)} - X^{(k)})C(Y_m^{(k)} - X^{(k)}). \tag{6}$$

2. Set $X^{(k+1)} = Y_r^{(k)}$ and form the vector $x^{(k+1)} = \alpha\Delta X^{(k+1)}e + e$.

Obviously, by letting

$$y_m^{(k)} = \alpha\Delta Y_m^{(k)}e + e, \tag{7}$$

the last item in iterative scheme (6) admits the following factorization:

$$(Y_m^{(k)} - X^{(k)})C(Y_m^{(k)} - X^{(k)}) = (\Delta^{-1}(y_m^{(k)} - x^{(k)}))(\Delta^{-1}(y_m^{(k)} - x^{(k)}))^T.$$

Then, rather than solving the equation of a rank-two matrix on RHS in structured Newton's method, the structured Shamanskii method copes with one of a rank-three matrix on RHS. Analogously, the factor-alternating direction implicit (FADI) method [2,16,20,22,23] might be applicable to the subproblem in each Shamanskii step. In details, set

$$F_{m1}^{(k)} = \Delta^{-1}[e,\, x_m^{(k)} - e,\, y_m^{(k)} - x^{(k)}], \quad G_{m1}^{(k)} = \Delta^{-1}[e,\, e - x_m^{(k)},\, y_m^{(k)} - x^{(k)}],$$

then each $Y_m^{(k)}$ (also $X^{(k)}$) in structured Shamanskii method has the following low-rank factorization when the number J_k of FADI iterations is determined:

$$\begin{cases} Y_m^{(k)} = T_m^{(k)}(\bar{T}_m^{(k)})^T, \\ T_m^{(k)} = [\, T_{m1}^{(k)}, \frac{\sqrt{2p_2^{(k)}}}{\sqrt{2p_1^{(k)}}}\,(p_1^{(k)}I - A + X^{(k)}C)(A - X^{(k)}C + p_2^{(k)}I)^{-1}T_{m1}^{(k)},\, ..., \\ \qquad \frac{\sqrt{2p_{J_k}^{(k)}}}{\sqrt{2p_{J_k-1}^{(k)}}}\,(p_{J_k-1}^{(k)}I - A + X^{(k)}C)(A - X^{(k)}C + p_{J_k}^{(k)}I)^{-1}T_{m,J_k-1}^{(k)}\,], \\ \bar{T}_m^{(k)} = [\, \bar{T}_{m1}^{(k)}, \frac{\sqrt{2p_2^{(k)}}}{\sqrt{2p_1^{(k)}}}\,(p_1^{(k)}I - A + X^{(k)}C)(A - X^{(k)}C + p_2^{(k)}I)^{-1}\bar{T}_{m1}^{(k)}, ..., \\ \qquad \frac{\sqrt{2p_{J_k}^{(k)}}}{\sqrt{2p_{J_k-1}^{(k)}}}\,(p_{J_k-1}^{(k)}I - A + X^{(k)}C)(A - X^{(k)}C + p_{J_k}^{(k)}I)^{-1}\bar{T}_{m,J_k-1}^{(k)}\,], \end{cases} \tag{8}$$

where

$$T_{m1}^{(k)} = \sqrt{2p_1^{(k)}}(A - X^{(k)}C + p_1^{(k)}I)^{-1}F_{m1}^{(k)},\ \bar{T}_{m1}^{(k)} = \sqrt{2p_1^{(k)}}(A - X^{(k)}C + p_1^{(k)}I)^{-1}G_{m1}^{(k)}$$

and

$$T_{mi}^{(k)} = \frac{\sqrt{2p_i^{(k)}}}{\sqrt{2p_{i-1}^{(k)}}}(p_{i-1}^{(k)}I - A + X^{(k)}C)(A - X^{(k)}C + p_i^{(k)}I)^{-1}T_{m,i-1}^{(k)},$$

$$\bar{T}_{mi}^{(k)} = \frac{\sqrt{2p_i^{(k)}}}{\sqrt{2p_{i-1}^{(k)}}}(p_{i-1}^{(k)}I - A + X^{(k)}C)(A - X^{(k)}C + p_i^{(k)}I)^{-1}\bar{T}_{m,i-1}^{(k)}$$

for $i = 2, ..., J_k - 1$. We refer to [2,16] for more details about the FADI iteration method.

The following theorem shows that the FADI iteration can be implemented in a more efficient format with the $O(n)$ computational complexity.

Theorem 1. *Let $T_m^{(k)}$ and $\bar{T}_m^{(k)}$ be factor matrices generated by the FADI iteration for solving (6). For each $i = 1, 2, ..., J_k$ and $m = 0, 1, ...r - 1$, denote by*

$$[t_{m,3i-2}^{(k)}, t_{m,3i-1}^{(k)}, t_{m,3i}^{(k)}] \;\; \text{and} \;\; [\bar{t}_{m,3i-2}^{(k)}, \bar{t}_{m,3i-1}^{(k)}, \bar{t}_{m,3i}^{(k)}]$$

the i-th triple vectors in $T_{mi}^{(k)}$ and $\bar{T}_{mi}^{(k)}$, respectively. Define diagonal matrices

$$D_{\Delta_1}^{(k)} = (I + p_1^{(k)}\Delta)^{-1}, \;\; D_{\Delta_i}^{(k)} = (p_{i-1}^{(k)}\Delta - I)(I + p_i^{(k)}\Delta)^{-1}, \;\; i = 2, 3, ..., J_k,$$

and scalars

$$\begin{aligned} &h_{m1}^{(k)} = \frac{\eta^T D_{\Delta_1}^{(k)} e}{1 - \eta^T D_{\Delta_1}^{(k)} x^{(k)}}, \; h_{m2}^{(k)} = \frac{\eta^T D_{\Delta_1}^{(k)}(x^{(k)} - e)}{1 - \eta^T D_{\Delta_1}^{(k)} x^{(k)}}, \; h_{m3}^{(k)} = \frac{\eta^T D_{\Delta_1}^{(k)}(y_m^{(k)} - x^{(k)})}{1 - \eta^T D_{\Delta_1}^{(k)} x^{(k)}}, \\ &h_{m,3i}^{(k)} = \frac{\eta^T (I + p_i\Delta)^{-1}\Delta t_{m,3i-3}^{(k)}}{1 - \eta^T (I + p_i\Delta)^{-1} x^{(k)}}, \; h_{m,3i-2}^{(k)} = \frac{\eta^T (I + p_i\Delta)^{-1}\Delta t_{m,3i-5}^{(k)}}{1 - \eta^T (I + p_i\Delta)^{-1} x^{(k)}}, \; h_{m,3i-1}^{(k)} = \frac{\eta^T (I + p_i\Delta)^{-1}\Delta t_{m,3i-4}^{(k)}}{1 - \eta^T (I + p_i\Delta)^{-1} x^{(k)}}, \end{aligned} \tag{9}$$

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

$$[\, t_{m1}^{(k)}, \, t_{m2}^{(k)}, \, t_{m3}^{(k)} \,] = \sqrt{2p_1}[D_{\Delta_1}^{(k)}(e + h_{m1}^{(k)} x^{(k)}), D_{\Delta_1}^{(k)}((1 + h_{m2}^{(k)})x^{(k)} - e), D_{\Delta_1}^{(k)}(y_m^{(k)} - x^{(k)} + h_{m3}^{(k)} x^{(k)})], \tag{10}$$

$$[\, \bar{t}_{m1}^{(k)}, \, \bar{t}_{m2}^{(k)}, \, \bar{t}_{m3}^{(k)} \,] = [t_{m1}^{(k)}, \, -t_{m2}^{(k)}, \, t_{m3}^{(k)}] \tag{11}$$

and for $i = 2, ..., J_k$

$$\begin{aligned} [\, t_{m,3i-2}^{(k)}, \, t_{m,3i-1}^{(k)}, \, t_{m,3i}^{(k)} \,] = \sqrt{2p_i} \; [\, &D_{\Delta_i}^{(k)}(t_{m,3i-5}^{(k)} + h_{m,3i-2}^{(k)}\xi^{(k)}) + h_{m,3i-2}^{(k)}\xi^{(k)}, \\ &D_{\Delta_i}^{(k)}(t_{m,3i-4}^{(k)} + h_{m,3i-1}^{(k)}\xi^{(k)}) + h_{m,3i-1}^{(k)}\xi^{(k)}, \\ &D_{\Delta_i}^{(k)}(t_{m,3i-3}^{(k)} + h_{m,3i}^{(k)}\xi^{(k)}) + h_{m,3i}^{(k)}\xi^{(k)} \,], \end{aligned} \tag{12}$$

$$[\, \bar{t}_{m,3i-2}^{(k)}, \, \bar{t}_{m,3i-1}^{(k)}, \, \bar{t}_{m,3i}^{(k)} \,] = [\, t_{3i-2}^{(k)}, \; -t_{m,3i-1}^{(k)}, \, t_{m,3i}^{(k)} \,] \tag{13}$$

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

Proof. By using the Sherman-Morrison-Woodbury formula (see [7] for example) for the inverse matrix of $A - X^{(k)}C + p_i^{(k)}I = \Delta^{-1} + p_i^{(k)}I - \xi^{(k)}\eta^T$ $(i = 1, ..., J_k)$ in (8), we can obtain the factoring formulae (10)–(13). □

Remark. (i). Apart from sharing the same scalars $h^{(k)}_{m,3i-2}$ and $h^{(k)}_{m,3i-1}$ and vectors $t^{(k)}_{m,3i-2}$ and $t^{(k)}_{m,3i-1}$ ($i = 1,2,...,J_k$) with structured Newton's method, the structured Shamanskii method additionally ushers in $h^{(k)}_{m,3i}$ and $t^{(k)}_{m,3i}$, which is exclusive of the m-th Shamanskii step. Therefore, scalars $h^{(k)}_{m,3i-2}$, $h^{(k)}_{m,3i-1}$ and vectors $t^{(k)}_{m,3i-2}$, $t^{(k)}_{m,3i-1}$ actually have no relation to m, the subscript imposed here is only for the purpose of the uniformity for whole iteration format.
(ii). Compared with structured Newton's method increasing two vectors within each iteration $T^{(k)}_m$ and $\bar{T}^{(k)}_m$, the structured Shamanskii method at m-step raises three instead. From the viewpoint of Krylov subspaces, the dimension of low-rank solution space at each iteration structured Shamanskii method enables to enhance $3r$ times, implying a faster convergence than that of the structured Newton's method.

We give all steps of the structured Newton-FADI iteration method for Chandrasekhar equation with $c \in [0,1]$ in Algorithm 1.

Algorithm 1. Inputs: Initial guess $x^{(0)} = 0$ and tolerance tol. Outputs: $x \in \mathbb{R}^n$ is approximative minimal positive solution of Chandrasekhar equation

1. $x^{(0)} := 0$.
2. For $k = 0,1,2,...$, until convergence, do
3. Compute the two extreme eigenvalues of $\Delta^{-1} - \Delta^{-1}(x^{(k)})\eta^T$.
4. Determine J_k and the optimal ADI parameters $\{p_i\}_{i=1}^{J_k}$.
5. Construct [$t^{(k)}_{01}$, $t^{(k)}_{02}$] via (10), store $1 - \eta^T(I + p_1\Delta)^{-1}x^{(k)}$ and $I + p_1\Delta$.
6. $y^{(0)} := ((t^{(k)}_{01})^T e)t^{(k)}_{01} - ((t^{(k)}_{02})^T e)t^{(k)}_{02}$.
7. For $i = 2,3,...,J_k$, do
8. Update $[h^{(k)}_{3i-2}, h^{(k)}_{3i-1}]$ with (9), $[t^{(k)}_{3i-2}, t^{(k)}_{3i-1}]$ with (12) and store $1 - \eta^T(I + p_i\Delta)^{-1}x^{(k)}$, $I + p_i\Delta$.
9. $y^{(k)}_0 := y^{(k)}_0 + ((t^{(k)}_{m,3i-2})^T e)t^{(k)}_{m,3i-2} - ((t^{(k)}_{m,3i-1})^T e)t^{(k)}_{m,3i-1}$.
10. End
11. $y^{(k)}_0 := \alpha\Delta y^{(k)}_0 + e$.
12. For $m = 0,...,r-1$, do
13. $v^{(k)} := y^{(k)}_m - x^{(k)}$. Update $h^{(k)}_{m3}$ with (9), $t^{(k)}_{m3}$ with (10) and set $y^{(k)}_{m+1} := (t_1^T e)t_1$.
17. For $i = 2,3,...,J_k$, do
18. Update $h^{(k)}_{m,3i}$ with (9), $t^{(k)}_{m,3i}$ with (12) and set $y^{(k)}_{m+1} := y^{(k)}_{m+1} + (t_1^T e)t_1$.
20. End
21. $y^{(k)}_{m+1} := y^{(k)}_0 + \alpha\Delta y^{(k)}_{m+1}$.
22. End
23. $x^{(k)} = y^{(k)}_r$.
24. If $||F(x^{(k)})||_2 < tol, x := x^{(k)}$, stop.
25. $k := k+1$.
26. End

2.2. Implementation issues

(i) For general implementations, the whole Algorithm 1 describes m-step structured Shamanskii method. But as stated in Introduction, the number $m = 2$ implies the efficiency index attaining the highest value. So, the numerical experiments in Section 4 indicate the performance of 2-step structured Shamanskii method.
(ii) The computation of two extremal eigenvalues of $\Delta^{-1} - \Delta^{-1}(x^{(k)})\eta^T$ and the determination of the J_k optimal ADI parameters $\{p_i\}$ in rows 3–4 are totally same with [24], see also [16,22,23] for more details.
(iii) Note that in row 8, the scalars $[h^{(k)}_{m,3i-2}, h^{(k)}_{m,3i-1}]$ of (9) and vectors $[t^{(k)}_{m,3i-2}, t^{(k)}_{m,3i-1}]$ of (12) are not relevant to m as stated in Remark (iii), we omit the subscript in Algorithm 1.
(iv) Recall that the structured Newton's method needs only rows 1–12 (except for row 9) with an additional

convergence detection row 24. While the structured Shamanskii method requires updating the vector $y_m^{(k)}$ extra mJ_k times (m = 2 as the efficiency index is the highest). Note that forming a J_k vector and an $n \times J_k$ matrix for respective storage of $1-\eta^T(I+p_i\Delta)^{-1}x^{(k)}$ and $I+p_i\Delta$ $(i=1,\ldots J_k)$ in row 9 is unavoidable as they are indispensible for the update of $h_{m,3i}^{(k)}$ and $t_{m,3i}^{(k)}$ in rows 15 and 18. Therefore, the structured Shamanskii method definitely requires more storage space compared to the structured Newton's version. Fortunately, the slightly higher storage cost could be ignored as faster convergence of Shamanskii method decreases CPU time.

3. CONVERGENCE AND ELEMENTARY MONOTONICITY

To show the monotone convergence of the structured Shamanskii method, a lemma is first required (see [8,9]).

Lemma 3. *Let X^* be the minimal nonnegative solution of the algebraic Riccati equaton* (3). *Suppose that a matrix X is such that*
(i) $\mathscr{R}(X) \geq 0$,
(ii) $0 \leq X \leq X^*$, *and* $0 \leq X < X^*$ *when K is an irreducible singular M-matrix*,
(iii) $I \otimes (A-XC) + (A-XC) \otimes I$ *is a nonsingular M-matrix.*
Then for any matrix Z with $0 \leq Z \leq X$, *the matrix*

$$Y = X - (\mathscr{R}_Z^{'})^{-1}\mathscr{R}(X) \tag{14}$$

i*s well defined and*
(a) $\mathscr{R}(Y) \geq 0$,
(b) $0 \leq Y \leq X^*$, *and* $0 \leq Y < X^*$ *when K is an irreducible singular M-matrix*,
(c) $I \otimes (A-YC) + (A-YC) \otimes I$ *is a nonsingular M-matrix.*

Theorem 2. *Let x^* be the minimal nonnegative solution of the Chandrasekhar equation* (1). *Then the structured Shamanskii method generates sequence $\{x^{(k)}\}$ and $\{y^{(k)}\}$ satisfying*

$$x^{(k)} = y_0^{(k)} \leq y_1^{(k)} \leq \ldots \leq y_r^{(k)} = x^{(k+1)} \leq x^*$$

for all $k \geq 0$ and $\lim_{k\to\infty} x^{(k)} = x^*$.

Proof. Starting with $x^{(0)} = 0$, the initial iteration matrix in structured Shamanskii method (6) is $Y_0^{(0)} = X^{(0)} = 0$.

Let $Z = X^{(0)}$, $Y = Y_1^{(0)}$ and $X = X^{(0)}$ in (14). Obviously, the conditions (i), (ii), and (iii) in Lemma 3 hold true and we have $\mathscr{R}(Y_1^{(0)}) \geq 0$, $Y_1^{(0)} \leq X^*$ and $I \otimes (A-Y_1^{(0)}C) + (A-Y_1^{(0)}C) \otimes I$ is a non-singular M-matrix. Then the equivalent form of iteration (14)

$$\Big(I \otimes (A-Y_1^{(0)}C) + (A-Y_1^{(0)}C) \otimes I\Big)\mathrm{vec}(Y_1^{(0)} - Y_0^{(0)}) = \mathrm{vec}(\mathscr{R}(Y_1^{(0)}))$$

implies $X^* \geq Y_1^{(0)} \geq Y_0^{(0)} = X^{(0)}$. If we assume that $X^* \geq Y_j^{(0)} \geq Y_{j-1}^{(0)} \geq X^{(0)}$ holds for some integer j, it is easy to validate that these inequalities also hold true for $j+1$ by letting $Z = X^{(0)}$, $Y = Y_{j+1}^{(0)}$ and $X = Y_j^{(0)}$ in (14). Thus we have shown

$$X^{(0)} = Y_0^{(0)} \leq Y_1^{(0)} \leq \ldots \leq Y_r^{(0)} = Y^{(1)} \leq X^*. \tag{15}$$

Now, suppose (15) is true for some integer $k-1$, i.e., $\mathscr{R}(X^{(k)}) = \mathscr{R}(Y_r^{(k-1)}) \geq 0$, $X^{(k)} = Y_r^{(k-1)} \leq X^*$, and $I \otimes (A-X^{(k)}C) + (A-X^{(k)}C) \otimes I = I \otimes (A-Y_r^{(k-1)}C) + (A-Y_r^{(k-1)}C) \otimes I$ is a nonsingular M-matrix, we shall indicate by induction that (15) is also valid for k. In fact, setting $X = X^{(k)}$ in Lemma 3 satisfies conditions (i), (ii) and (iii), and letting $Z = X^{(k)}$, $Y = Y_1^{(k)}$. We then have $\mathscr{R}(Y_1^{(k)}) \geq 0$ and $Y_1^{(k)} \leq X^*$. Moreover $I \otimes (A - Y_1^{(k)}C) + (A - Y_1^{(k)}C) \otimes I$ is a nonsingular M-matrix. Repeating the same argument with case $k = 0$, it holds $X^* \geq Y_1^{(k)} \geq Y_0^{(k)} = X^{(k)}$. Again, an induction argument of $X^* \geq Y_j^{(k)} \geq Y_{j-1}^{(k)} = X^{(k)}$ for $j = 1, ..., r$ shows that (15) is valid for k.

Therefore we know that the matrix sequence $\{X^{(k)}\}$ is monotonically increasing and bounded above by X^*. Then it has a limit $\bar{X}^*$ satisfying the Eq. (3), i.e. $\bar{X}^*$ is a nonnegative solution of (3). As $\bar{X}^* \leq X^*$, $\bar{X}^*$ is the minimal nonnegative solution X^*. Note the constitution of $y_m^{(k)}$ and $x_m^{(k)}$ in structured Shamanskii method, the conclusion is drawn in Theorem 2. □

Theorem 3. *Let the sequence* $\{y_m^{(k)}\}_{k=1}^{+\infty}$ $(m = 0, 1, ..., r)$ *be generated by structured Shamnaskii method with an initial guess* $x^{(0)} = y^{(0)} = 0$. *Then for* $k = 1, 2, \ldots$, *each iteration vector* $y_m^{(k)}$ *is elementwisely strictly monotonic, namely, for* $m = 0, 1, ..., r$ *and* $k = 1, 2, \ldots$, *the following inequalities hold true*

$$(y_m^{(k)})_n \geq (y_m^{(k)})_{n-1} \geq ... \geq (y_m^{(k)})_2 \geq (y_m^{(k)})_1. \tag{16}$$

Proof. The iteration format of the structured Shamanskii method could be written as

$$\begin{cases} y_0^{(k)} = x^{(k)}, \\ y_{m+1}^{(k)} = (M^{(k)})^{-1}\Big((y_m^{(k)} - x^{(k)}) \circ (S(y_m^{(k)} - x^{(k)})) + e - x^{(k)} \circ Sx^{(k)}\Big), \; m \geq 0, \\ x^{(k+1)} = y_r^{(k)}, \end{cases}$$

where $M^{(k)} = I - \mathrm{diag}(Sx^{(k)}) - \mathrm{diag}(x^{(k)})S$ and the elements of S are $(S)_{ij} = \alpha \frac{\delta_i}{\delta_i + \delta_j}$. Note that the monotonicity of $\delta_1 < ... < \delta_n$, it always holds $(Sv)_{i+1} > (Sv)_i$ for any vector $v \geq 0$ and $v \neq 0$. On the other hand, it follows an analogous way in [8, Thm 3.1] that $A - X^{(k)}C = \Delta^{-1} - \alpha\Delta^{-1}x^{(k)}e^T$ is a nonsingular M-matrix, so is $I - \alpha x^{(k)}e^T$, which further implies for any $i = 1, ..., n$ and $k = 0, 1, ...,$

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

Then $e > Sx^{(k)}$ holds true for each $x^{(k)}$ in Shamanskii method.

We next demonstrate Theorem 3 by the induction. Starting with $x^{(0)} = 0$, it is clear $y_1^{(0)} = e$ and (16) holds for $k = 0$ and $m = 1$. Suppose (16) is true for $k = 0$ and $m = j$. Then

$$(y_{j+1}^{(0)})_i = (y_j^{(0)})_i (Sy_j^{(0)})_i \leq (y_j^{(0)})_{i+1} (Sy_j^{(0)})_{i+1} = (y_{j+1}^{(0)})_{i+1}$$

with $i = 1, ..., n-1$ indicates (16) holds for $k = 0$ and $m = j+1$. So the inequality (16) is valid for $k = 0$ and all $m = 1, 2, ...r$. Now assume (16) is true for $k = l$ and $m = 1, 2, ...r$, we have to show it holds for $k = l+1$ and $m = 1, 2, ...r$. In fact, the inequality

$$(y_1^{(l+1)})_i = \frac{1 + (x^{(l+1)})_i (S(y_1^{(l+1)} - x^{(l+1)}))_i}{1 - (Sx^{(l+1)})_i} \leq \frac{1 + (x^{(l+1)})_{i+1} (S(y_1^{(l+1)} - x^{(l+1)}))_{i+1}}{1 - (Sx^{(l+1)})_{i+1}} = (y_1^{(l+1)})_{i+1}$$

indicates (16) is true for $k = l+1$ and $m = 1$ and, with assuming the validation of $m = 1, 2, ...j$, the inequality

$$\begin{aligned} (y_{j+1}^{(l+1)})_i &= \frac{1 + (x^{(l+1)})_i (S(y_{j+1}^{(l+1)} - x^{(l+1)}))_i}{1 - (Sx^{(l+1)})_i} + \frac{(y_j^{(l+1)} - x^{(l+1)})_i (S(y_j^{(l+1)} - x^{(l+1)}))_i}{1 - (Sx^{(l+1)})_i} \\ &\leq \frac{1 + (x^{(l+1)})_{i+1} (S(y_{j+1}^{(l+1)} - x^{(l+1)}))_{i+1}}{1 - (Sx^{(l+1)})_{i+1}} + \frac{(y_j^{(l+1)} - x^{(l+1)})_{i+1} (S(y_j^{(l+1)} - x^{(l+1)}))_{i+1}}{1 - (Sx^{(l+1)})_{i+1}} \\ &= (y_{j+1}^{(l+1)})_{i+1} \end{aligned}$$

shows (16) is true for $k = l+1$ and $m = j+1$. Thus the whole proof completes by induction. □

4. NUMERICAL EXPERIMENTS

This section will show the effectiveness of the structured Shamanskii method. The numerical example comes from the discrete integral equation (1) with different $c \in [0,1]$, see [13] for more details. In practical implementations, Algorithm 1 with one Shamanskii step (i.e. $r = 1$) and two Shamanskii steps (i.e. $r = 2$) is denoted by "SS1" and "SS2", respectively. Higher Shamanskii steps are omitted as they give poorer numerical results. In comparison, the structured Newton's method ("SN") in [20,24] is employed to highlight the efficiency of Algorithm 1.

All algorithms are coded by MATLAB 2014 and run on a PC with Intel i3-3240 3.4GHz processor and 8GB RAM. Besides, they are terminated when the inequality

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

is satisfied, where τ_r and τ_a are, relative error tolerance and absolute error tolerance, respectively [13]; and are set to be 10^{-12} in our experiments. The Chandrasekhar equation (1) varies from the non-critical case to extremely near-critical case via setting $c = 0.5$, 0.9, 0.99 and 0.999999. Numerical experiments are implemented in all cases and the results are listed in Tables 1–2, where the "n" column is the size of the problem, the "CPU" row denotes the CPU time used in seconds, the "IT" row represents the number of iterations, and the "RES" row stands for the 2-norm of the relative residual of the Chandrasekhar equation at the obtained minimal positive solution.

We see from Tables 1–2 that all three methods stopped regularly and were able to derive the minimal positive solution of (1). Indeed, the structured Shamanskii method performs better than the structured Newton's method in most cases.

Chandrasekhar equation (1) lies in the non-critical case when $c = 0.5$ and numerical results in Table 1 show that "SS1" and "SS2" are able to attain almost the same RES accuracy by using less CPU time with smaller iteration number. Especially, the "SS2" requires the shortest CPU time compared to others and thrift with CPU time becomes more obvious as the dimension n increases. When the equation is close to the critical case with parameter $c = 0.9$, an interesting numerical phenomenon can be seen in Table 1, where the "SS1" fails to obtain higher RES accuracy, although it beats "SN" in both – aspects of CPU time and iteration number. Still, the "SS2" outperforms other two in terms of CPU time without sacrificing the RES accuracy.

Table 2 shows the superiority of the structured Shamanskii method over Newton's method when the Chandasekhar equation approaches the critical case. Indeed, "SS1" and "SS2" achieve almost the same RES accuracy with that of "SN", but they are observed to consume less CPU time with smaller iteration number. Again, the "SS2" possesses the best numerical performance among all three, expending the shortest CPU time with the smallest iteration number. When the Chandrasekhar equation is extremely close to the critical case, the "SS2" requires only nearly half of the iteration number of the "SN" to attain the prescribed residual accuracy.

5. CONCLUSIONS

A structured Shamanskii method was presented in this paper to solve Chandrasekhar equation. The proposed method sufficiently exploits the low-cost chord iterations to accelerate the structured Newton's method and is still proved to possess the monotone convergence. Numerical experiments indicate that the proposed structured Shamanskii method with $r = 2$ outperforms structured Shamanskii method with $r = 1$ and the structured Newton's method. Hopefully, similar line of research will be conducted to generalize the newly developed method for more complicated Chandrasekhar equations presented in [11].

Table 1. Test results for $c = 0.5$ and $c = 0.9$

		$c = 0.5$			$c = 0.9$		
n	Method	SN	SS1	SS2	SN	SS1	SS2
5000	CPU	0.390	0.312	0.312	0.500	0.234	0.343
	IT	4	3	2	5	3	3
	RES	3.49e-14	3.49e-14	3.55e-14	1.34e-13	1.53e-11	1.34e-13
6000	CPU	0.500	0.390	0.296	0.500	0.312	0.375
	IT	4	3	2	5	3	3
	RES	4.19e-14	4.20e-12	4.21e-14	1.58e-13	1.68e-11	1.58e-13
7000	CPU	0.578	0.437	0.312	0.07	0.04	0.10
	IT	4	3	2	5	3	3
	RES	4.90e-14	4.92e-14	4.90e-14	1.88e-13	1.81e-11	1.88e-13
8000	CPU	0.859	0.703	0.687	0.718	0.484	0.468
	IT	4	3	2	5	3	3
	RES	5.46e-14	5.46e-14	5.49e-14	2.09e-13	1.94e-11	2.09e-13
9000	CPU	0.968	0.640	0.531	1.296	0.796	0.828
	IT	4	3	2	5	3	3
	RES	6.19e-14	6.19e-14	6.20e-14	2.37e-13	2.05e-11	2.37e-13
10000	CPU	1.062	0.875	0.734	1.171	0.890	0.937
	IT	4	3	2	5	3	3
	RES	6.86e-14	6.86e-14	6.88e-14	5.73e-13	2.84e-11	6.19e-13
11000	CPU	1.218	1.000	0.718	1.750	0.968	0.968
	IT	4	3	2	5	3	3
	RES	7.49e-14	7.48e-14	7.53e-14	2.89e-13	2.77e-11	2.89e-13
12000	CPU	1.984	1.406	0.859	1.734	1.265	1.688
	IT	4	3	2	5	3	3
	RES	7.83e-15	2.22e-12	5.98e-14	3.22e-13	2.37e-11	3.23e-13
13000	CPU	1.734	1.359	1.109	1.937	1.234	1.312
	IT	4	3	2	5	3	3
	RES	8.96e-14	8.98e-14	9.01e-14	3.44e-13	2.47e-11	3.44e-13
14000	CPU	2.093	1.750	1.221	2.156	1.515	1.484
	IT	4	3	2	5	3	3
	RES	9.61e-14	9.60e-14	9.62e-14	3.79e-13	2.56e-11	3.79e-13
15000	CPU	2.265	1.781	1.250	2.718	2.406	1.875
	IT	4	3	2	5	3	3
	RES	1.03e-13	1.03e-13	1.03e-13	4.08e-13	2.65e-11	4.07e-13
16000	CPU	3.171	2.109	1.515	3.171	2.078	2.040
	IT	4	3	2	5	3	3
	RES	1.08e-13	1.09e-13	1.09e-13	4.28e-13	2.74e-11	4.27e-13

Table 2. Test results for $c = 0.99$ and $c = 0.999999$

		$c = 0.99$			$c = 0.999999$		
n	Method	SN	SS1	SS2	SN	SS1	SS2
5000	CPU	0.500	0.453	0.390	0.906	0.906	0.750
	IT	7	5	4	13	9	7
	RES	2.68e-13	2.69e-13	2.67e-13	1.78e-11	1.79e-11	1.79e-11
6000	CPU	0.703	0.640	0.515	1.281	1.125	0.890
	IT	7	5	4	13	9	7
	RES	3.23e-13	3.22e-13	3.22e-13	1.94e-11	1.91e-11	1.91e-11
7000	CPU	1.234	1.328	0.796	1.546	1.390	1.234
	IT	7	5	4	13	9	7
	RES	3.77e-13	3.79e-13	3.77e-13	2.04e-11	2.05e-11	2.04e-11
8000	CPU	1.218	0.843	0.828	2.062	1.796	1.281
	IT	7	5	4	13	9	7
	RES	4.29e-13	4.30e-13	4.27e-13	2.25e-11	2.24e-11	2.18e-11
9000	CPU	1.781	1.375	1.093	2.437	2.265	1.640
	IT	7	5	4	13	9	7
	RES	4.88e-13	4.88e-13	4.89e-13	2.34e-11	2.35e-11	2.32e-11
10000	CPU	1.609	1.250	1.109	3.984	2.984	2.562
	IT	7	5	4	13	9	7
	RES	5.375e-13	5.35e-13	5.22e-13	2.46e-11	2.48e-11	2.46e-11
11000	CPU	2.453	1.546	1.312	3.406	2.765	3.343
	IT	7	5	4	13	9	7
	RES	5.91e-13	5.92e-13	5.92e-13	2.66e-11	2.65e-11	2.63e-11
12000	CPU	2.843	2.281	1.812	5.203	4.093	3.015
	IT	7	5	4	13	9	7
	RES	6.43e-13	6.42e-13	6.43e-13	2.70e-11	2.69e-11	2.70e-11
13000	CPU	2.625	1.968	2.156	4.984	3.328	2.812
	IT	7	5	4	13	9	7
	RES	7.08e-13	7.07e-13	7.09e-13	2.83e-11	2.84e-11	2.81e-11
14000	CPU	3.171	2.734	1.890	4.984	3.984	3.218
	IT	7	5	4	13	9	7
	RES	7.67e-13	7.69e-13	7.70e-13	2.99e-11	2.98e-11	2.99e-11
15000	CPU	3.531	2.765	2.453	6.859	4.921	3.578
	IT	7	5	4	13	9	7
	RES	8.02e-13	8.01e-13	8.02e-13	3.08e-11	3.10e-11	3.05e-11
16000	CPU	4.296	3.156	2.500	7.321	6.515	4.500
	IT	7	5	4	13	9	7
	RES	8.60e-13	8.58e-13	8.60e-13	3.12e-11	3.14e-11	3.11e-11

ACKNOWLEDGEMENTS

This work was supported partly by the NSF of China (11801163) and NSF of Hunan Province (2018JJ4062, 2020JJ2071) and CSC (201908430048). The publication costs of this article were partially covered by the Estonian Academy of Sciences.

REFERENCES

1. Bini, D. A., Iannazzo, B., and Poloni, F. A Fast Newton's Method for a Nonsymmetric Algebraic Riccati Equation. *SIAM J. Matrix Anal. Appl.*, 2008, **30**(1), 276–290.
2. Benner, P., Li, R. C., and Truhar, N. On the ADI method for Sylvester equations. *J. Comp. Appl. Math.*, 2009, **233**, 1035–1045.
3. Chandrasekhar, S. *Radiative transfer*. Dover, New York, 1960.
4. Chandrasekhar, S. *Radiative transfer*. Courier Corporation, 2013.
5. Deeba, E. Y. and Khuri, S. A. The decomposition method applied to Chandrasekhar H-equation. *Appl. Math. Comput.*, 1996, **1**, 67–78.
6. Dong, N. and Yu, B. On the tripling algorithm for large-scale nonlinear matrix equations with low rank structure. *J. Comput. Appl. Math.*, 2015, **288**, 18–32.
7. Golub, G. H. and Van Loan C. F. *Matrix computations*, 3rd Edition. Johns Hopkins University Press, Baltimore, MD, 1996.
8. Guo, C.-H. Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices. *SIAM J. Matrix Anal. Appl.*, 2001, **23**, 225–242.
9. Guo, C.-H. Monotone convergence of Newton-like methods for M-matrix algebraic Riccati equations. *Numer. Algor.*, 2013, **64**, 295–309.
10. Guo, C.-H. and Laub, A. J. On the Iterative Solution of a Class of Nonsymmetric Algebraic Riccati Equations. *SIAM J. Matrix Anal. Appl.*, 2000, **22**(2), 376–391.
11. Juang, J. On Coupled Integral H-Like Equations of Chandrasekhar. *SIAM J. Math. Anal.*, 1995, **4**, 869–879.
12. Juang, J. and Lin, W.-W. Nonsymmetric Algebraic Riccati Equations and Hamiltonian-like Matrices. *SIAM J. Matrix Anal. Appl.*, 1998, **20**(1), 228–243.
13. Kelley, C. T. *Solving nonlinear equations with Newton's method*. Society for lndustrial and Applied Mathematics, Philadelphia, 2003.
14. Kelley, C. T., Kevrekidis, I. G., and Qiao, L. Newton-Krylov solvers for time-steppers. *Tech. Rep. CRSC-TR04-10, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC*, 2004.
15. Khuri, S. A. On the solution of coupled H-like equations of Chandrasekhar. *Appl. Math. Comput.*, 2002, **2-3**, 479–485.
16. Li, J.-R. and White, J. Low-rank Solution of Lyapunov Equations. *SIAM J. Matrix Anal. Appl.*, 2002, **24**(1), 260–280.
17. Li, R.-C. Solving secular equations stably and efficiently. Technical Report, *Department of Mathematics, University of California*, Berkeley, CA, LAPACK Working Note 1993, **89**. http://www2.eecs.berkeley.edu/Pubs/TechRpts/1994/CSD-94-851.pdf
18. Mehrmann, V. and Xu, H.-G. Explicit Solutions for a Riccati Equation from Transport Theory. *SIAM J. Matrix Anal. Appl.*, 2008, **30**(4), 1339–1357.
19. Traub, J. F. *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs, 1964.
20. Benner, P., Kurschner, P., and Saak, J. Low-rank Newton-ADI methods for large nonsymmtric algebraic Riccati euqaitons. *J. Franklin Instit.*, 2016, **353**(5), 1147–1167.
21. Varga, R. *Matrix iterative analysis*, 2nd Edition. Springer-Verlag, Berlin, Heidelberg, 2000.
22. Wachspress, E. L. *The ADI model problem*, Windsor, CA, 1995.
23. Wachspress, E. L. ADI iteration parameters for solving Lyapunov and Sylvester equations. *Technical Report*, March, 2009.
24. Yu, B., Li, D.-H., and Dong, N. Low memory and low complexity iterative schemes for a nonsymmetric algebraic Riccati equation arising from transport theory. *J. Comput. Appl. Math.*, 2013, **250**, 175–189.

Struktureeritud Šamanski meetodid radiatsioonis esineva Chandrasekhari võrrandi jaoks

Ning Dong, Bo Yu ja Zhaoyun Meng

On välja pakutud struktureeritud Šamanski meetod Chandrasekhari võrrandi minimaalse positiivse lahendi leidmiseks. On tõestatud meetodi monotoonne koonduvus. Numbriliste näidete baasil on näidatud, et antud meetod on struktureeritud Newtoni meetodist efektiivsem.