SIAM J. MATRIX ANAL. APPL.
Vol. 34, No. 3, pp. 1129–1147

SOLVING LARGE-SCALE NONSYMMETRIC ALGEBRAIC RICCATI EQUATIONS BY DOUBLING[*]

TIEXIANG LI[†], ERIC KING-WAH CHU[‡], YUEH-CHENG KUO[§], AND WEN-WEI LIN[¶]

Abstract. We consider the solution of the large-scale nonsymmetric algebraic Riccati equation $XCX - XD - AX + B = 0$, with $M \equiv [D, -C; -B, A] \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ being a nonsingular M-matrix. In addition, A and D are sparselike, with the products $A^{-1}u$, $A^{-\top}u$, $D^{-1}v$, and $D^{-\top}v$ computable in $O(n)$ complexity (with $n = \max\{n_1, n_2\}$), for some vectors u and v, and B, C are low ranked. The structure-preserving doubling algorithms (SDA) by Guo, Lin, and Xu [*Numer. Math.*, 103 (2006), pp. 392–412] is adapted, with the appropriate applications of the Sherman–Morrison–Woodbury formula and the sparse-plus-low-rank representations of various iterates. The resulting large-scale doubling algorithm has an $O(n)$ computational complexity and memory requirement per iteration and converges essentially quadratically. A detailed error analysis, on the effects of truncation of iterates with an explicit forward error bound for the approximate solution from the SDA, and some numerical results will be presented.

Key words. doubling algorithm, M-matrix, nonsymmetric algebraic Riccati equation, numerically low-ranked solution

AMS subject classifications. 15A24, 65F50

DOI. 10.1137/110858070

1. Introduction.

Consider the nonsymmetric algebraic Riccati equation (NARE)

$$\mathcal{R}(X) \equiv XCX - XD - AX + B = 0, \tag{1.1}$$

where A, B, C, and D are real $n_1 \times n_1$, $n_1 \times n_2$, $n_2 \times n_1$, and $n_2 \times n_2$ matrices, respectively. From the solvability conditions in [12, 13], we assume the matrix

$$M = \begin{bmatrix} D & -C \\ -B & A \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)} \tag{1.2}$$

is a nonsingular M-matrix, i.e., M has nonpositive off-diagonal entries and all elements of M^{-1} are nonnegative. In this paper, we are interested in developing an efficient algorithm for solving the minimal nonnegative solution X of NAREs in (1.1).

The structure-preserving doubling algorithm (SDA) in [17] is first proposed for solving the NARE (1.1) with quadratical convergence. Then in [5], more general convergence results were given, especially for the critical case. Later in [4], Bini,

[*] Received by the editors December 7, 2011; accepted for publication (in revised form) by R.-C. Li May 29, 2013; published electronically July 30, 2013. The research of the third and the fourth authors was supported from the National Science Council and the National Centre for Theoretical Sciences in Taiwan.

http://www.siam.org/journals/simax/34-3/85807.html

[†] Corresponding author. Department of Mathematics, Southeast University, Nanjing 211189, People's Republic of China (txli@seu.edu.cn). The research of this author was supported by the NSFC (11101080) and the SRFDP (20110092120023), China.

[‡] School of Mathematical Sciences, Monash University, Melbourne, 3800, Australia (eric.chu@monash.edu).

[§] Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung 811, Taiwan (yckuo@nuk.edu.tw).

[¶] Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan (wwlin@math.nctu.edu.tw). The research of this author was supported by the CMMSC and the ST Yau Centre at the National Chiao Tung University.

Meini, and Poloni developed a doubling algorithm called SDA_ss, which has shown efficient improvements over SDA in some numerical tests, but it can happen that sometimes SDA_ss runs slower than SDA. Recently in [31], the alternating-directional doubling algorithm (ADDA) has been developed by Wang, Wang, and Li to improve the convergence of the SDA dramatically. In practice, ADDA is always faster than SDA and SDA_ss, however, it may encounter overflow in F_k and E_k before H_k and G_k converge with the desired accuracy, and the scaling technique in [31] is not suitable for the large-scale case which we will study.

We state the SDA for solving (1.1) as follows. Choose a suitable parameter γ such that

$$\gamma \geq \gamma_0 \equiv \max\left\{ \max_{1\leq i\leq n_1} a_{ii}, \max_{1\leq i\leq n_2} d_{ii} \right\}, \tag{1.3}$$

where a_{ii} and d_{ii} are the diagonal entries of A and D, respectively. Compute

$$\begin{aligned} F_0 &= I_{n_1} - 2\gamma W_\gamma^{-1}, \quad & E_0 &= I_{n_2} - 2\gamma V_\gamma^{-1}, \\ H_0 &= 2\gamma W_\gamma^{-1} B D_\gamma^{-1}, \quad & G_0 &= 2\gamma D_\gamma^{-1} C W_\gamma^{-1} \end{aligned} \tag{1.4}$$

with $A_\gamma \equiv A + \gamma I_{n_1}$, $D_\gamma \equiv D + \gamma I_{n_2}$, $W_\gamma \equiv A_\gamma - BD_\gamma^{-1}C$, $V_\gamma \equiv D_\gamma - CA_\gamma^{-1}B$. The SDA [17] has the form (for $k \geq 0$)

$$\begin{aligned} F_{k+1} &= F_k(I_{n_1} - H_kG_k)^{-1}F_k, & E_{k+1} &= E_k(I_{n_2} - G_kH_k)^{-1}E_k, \\ H_{k+1} &= H_k + F_k(I_{n_1} - H_kG_k)^{-1}H_kE_k, \quad & G_{k+1} &= G_k + E_k(I_{n_2} - G_kH_k)^{-1}G_kF_k, \end{aligned} \tag{1.5}$$

where $F_k \in \mathbb{R}^{n_1\times n_1}$, $E_k \in \mathbb{R}^{n_2\times n_2}$, $H_k \in \mathbb{R}^{n_1\times n_2}$, and $G_k \in \mathbb{R}^{n_2\times n_1}$.

For $A = [a_{ij}]$, $B = [b_{ij}] \in \mathbb{R}^{m\times n}$, we write $A \geq B$ $(A > B)$ if $a_{ij} \geq b_{ij}$ $(a_{ij} > b_{ij})$ for all i, j. A matrix A is called positive (nonnegative) if $a_{ij} > 0$ $(a_{ij} \geqslant 0)$. We denote $|A| = [|a_{ij}|]$ and $\|A\| := \|A\|_2$ the 2-norm of A.

Let

$$\mathcal{D}(Y) \equiv YBY - YA - DY + C = 0 \tag{1.6}$$

be the dual equation of NARE (1.1). The following convergence theorem for (1.5) is originally given in [17] and improved in [5].

Theorem 1.1. *Let M in* (1.2) *be a nonsingular M-matrix. Then the NARE* (1.1) *and its dual equation* (1.6) *have minimal nonnegative solutions $X \geq 0$ and $Y \geq 0$, respectively. Moreover $S = A - BY$ and $R = D - CX$ are nonsingular M-matrices. Let $S_\gamma = (S + \gamma I_{n_1})^{-1}(S - \gamma I_{n_1})$ and $R_\gamma = (R + \gamma I_{n_2})^{-1}(R - \gamma I_{n_2})$. Then the sequences $\{F_k\}$, $\{E_k\}$, $\{H_k\}$, and $\{G_k\}$ generated by the SDA* (1.5) *are well-defined, and for all $k \geq 0$, we have*

(a) *E_0, $F_0 \leq 0$, and $F_k = (I_{n_1} - H_kY)S_\gamma^{2^k} \geq 0$, $E_k = (I_{n_2} - G_kX)R_\gamma^{2^k} \geq 0$;*
(b) *$I - G_kH_k$ and $I - H_kG_k$ are nonsingular M-matrices;*
(c) *$0 \leq H_k \leq H_{k+1} \leq X$ and $0 \leq X - H_k = (I_{n_1} - H_kY)S_\gamma^{2^k} X R_\gamma^{2^k} \leq S_\gamma^{2^k} X R_\gamma^{2^k}$;*
(d) *$0 \leq G_k \leq G_{k+1} \leq Y$ and $0 \leq Y - G_k = (I_{n_2} - G_kX)R_\gamma^{2^k} Y S_\gamma^{2^k} \leq R_\gamma^{2^k} Y S_\gamma^{2^k}$;*
(e) *$S_\gamma, R_\gamma \leq 0$, the spectral radii $\rho(S_\gamma), \rho(R_\gamma) < 1$, and $S_\gamma^{2^k}, R_\gamma^{2^k} \to 0$ as $k \to \infty$;*
(f) *$I_{n_1} - XY$ and $I_{n_2} - YX$ are nonsingular M-matrices.*

Motivated by the low-ranked cases from the applications in transport theory [19, 22, 23], in this paper we further assume that n_1 and n_2 are large, A and D are sparselike (with the products $A^{-1}u$, $A^{-\top}u$, $D^{-1}v$, and $D^{-\top}v$ computable in $O(n)$

complexity, where $n = \max\{n_1, n_2\}$, for some vectors u and v), and B and C are low ranked (m and l, respectively, with $m, l \ll n_1, n_2$). In [19, 22, 23], A and D are low-ranked updates of nonsingular diagonal matrices, which are nonsingular but not sparse. We shall adapt the SDA [17] to solve the NARE (1.1), resulting in a large-scale doubling algorithm (SDA_ls_ε) with an $O(n)$ computational complexity and memory requirement per iteration. Note that the orthodox SDA in [17] has a computational complexity of $O(n^3)$.

More generally, algebraic Riccati equations arise in many important applications, including the total least squares problems with or without symmetric constraints [9], the spectral factorizations of rational matrix functions [10], the linear and nonlinear optimal controls [2], the contractive rational matrix functions [20], the structured complex stability radius [18], transport theory [19, 22, 23], the Wiener–Hopf factorization of Markov chains [32], and the optimal solutions of linear differential systems [21]. Symmetric algebraic Riccati equations have been the topic of extensive research, and the theory, applications, and numerical solutions of these equations are the subject of [5, 6, 7, 8] as well as the monographs [21, 29]. The minimal positive solution to the NARE (1.1), for medium size problems without the sparseness and low-ranked assumptions, has been studied recently by several authors, employing functional iterations, Newton's method, and the structure-preserving algorithm; see [1, 3, 4, 12, 13, 14, 15, 16, 17, 22, 23, 26, 27, 30, 31], and the references therein. Evidently, the applications associated with and the numerical solution to NAREs have attracted much attention in the past decade but this paper is the first on general large-scale NAREs.

Main contributions. Apart from being the first paper on the numerical solution to general large-scale NAREs, we shall formalize the discussion on the numerical rank of the solution X, showing constructively when X is numerically low ranked. We adapt the well-known structure-preserving doubling method efficiently for large-scale NAREs. Then we show how the exponential growth in the rank of the approximate solution is controlled by compression and truncation. A first order error estimate will show that the difference between the approximate solution by the SDA with small truncation and the exact solution has the same order as the truncation without affecting the convergence of the SDA.

2. Large-scale doubling algorithm.

Borrowing from [24], we shall apply the Sherman–Morrison–Woodbury formula (SMWF) in order to avoid the inversion of large or unstructured matrices, and use sparse-plus-low-ranked matrices to represent iterates when appropriate. Also, some matrix operators are computed recursively, to preserve the corresponding sparsity or low-ranked structures, instead of forming them explicitly. If necessary, we compress and truncate fast growing components in the iterates, to trade off the negligible amount of accuracy for better efficiency. Together with the careful organization of convergence control in the algorithm, we obtain an $O(n)$ computational complexity and memory requirement per iteration.

2.1. Large-scale SDA.

We assume $B \in \mathbb{R}^{n_1 \times n_2}$ and $C \in \mathbb{R}^{n_2 \times n_1}$ in (1.1) have, respectively, the full low-ranked decompositions

$$B = B_1 B_2^\top, \quad C = C_1 C_2^\top, \tag{2.1}$$

where $B_1 \in \mathbb{R}^{n_1 \times m}, B_2 \in \mathbb{R}^{n_2 \times m}, C_1 \in \mathbb{R}^{n_2 \times \ell}, C_2 \in \mathbb{R}^{n_1 \times \ell}$ with $m,\ l \ll n \equiv \max\{n_1, n_2\}$. We first state a basic large-scale SDA, and then propose a practical large-scale SDA later in section 2.2.

For the initial matrices in (1.4), we have $F_0 = I_{n_1} - 2\gamma W_\gamma^{-1}$, $E_0 = I_{n_2} - 2\gamma V_\gamma^{-1}$, $H_0 = Q_{10}\Sigma_0 Q_{20}^\top$, and $G_0 = P_{10}\Gamma_0 P_{20}^\top$, where

$$\begin{aligned} &Q_{10} \equiv 2\gamma W_\gamma^{-1} B_1, \quad Q_{20} \equiv D_\gamma^{-\top} B_2, \quad \Sigma_0 \equiv I_m, \\ &P_{10} \equiv 2\gamma D_\gamma^{-1} C_1, \quad P_{20} \equiv W_\gamma^{-\top} C_2, \quad \Gamma_0 \equiv I_l. \end{aligned} \tag{2.2}$$

Note that efficient linear solvers for the large scale A and D, and thus for A_γ and D_γ, are available. Applying the SMWF, $W_\gamma^{-1} w$ and $V_\gamma^{-1} v$ can be computed economically by

$$W_\gamma^{-1} w = \left\{ I_{n_1} + A_\gamma^{-1} B_1 \left[I_m - (B_2^\top D_\gamma^{-1} C_1)(C_2^\top A_\gamma^{-1} B_1) \right]^{-1} (B_2^\top D_\gamma^{-1} C_1) C_2^\top \right\} A_\gamma^{-1} w, \tag{2.3}$$

$$V_\gamma^{-1} v = \left\{ I_{n_2} + D_\gamma^{-1} C_1 \left[I_l - (C_2^\top A_\gamma^{-1} B_1)(B_2^\top D_\gamma^{-1} C_1) \right]^{-1} (C_2^\top A_\gamma^{-1} B_1) B_2^\top \right\} D_\gamma^{-1} v. \tag{2.4}$$

For $k = 1, 2, \ldots$, we shall organize the SDA so that the iterates have the recursive forms

$$H_k = Q_{1k}\Sigma_k Q_{2k}^\top, \qquad G_k = P_{1k}\Gamma_k P_{2k}^\top, \tag{2.5}$$

$$F_k = F_{k-1}^2 + F_{1k}F_{2k}^\top, \quad E_k = E_{k-1}^2 + E_{1k}E_{2k}^\top, \tag{2.6}$$

where $F_{ik} \in \mathbb{R}^{n_1 \times l_{k-1}}$, $E_{ik} \in \mathbb{R}^{n_2 \times m_{k-1}}$ $(i = 1, 2)$, and the kernels $\Sigma_k \in \mathbb{R}^{m_k \times m_k}$ and $\Gamma_k \in \mathbb{R}^{l_k \times l_k}$.

We should compute products like $E_k u$, $E_k^\top u$, $F_k v$, $F_k^\top v$, for some vectors u and v, by applying (2.6) recursively. Without actually forming E_k and F_k, we avoid any possible deterioration of their sparselike properties or other structures as k grows, and preserve the $O(n)$ computational complexity of the algorithm. As a trade-off, we need to store all the Q_{ik}, Σ_k, P_{ik}, Γ_k, F_{ik}, and E_{ik} for all previous k, as we shall see below.

Applying the SMWF again, we obtain

$$\begin{aligned} (I_{n_2} - G_k H_k)^{-1} &= I_{n_2} + G_k Q_{1k}\Sigma_k \left(I_{m_k} - Q_{2k}^\top G_k Q_{1k}\Sigma_k \right)^{-1} Q_{2k}^\top \\ &= I_{n_2} + P_{1k} \left(I_{l_k} - \Gamma_k P_{2k}^\top H_k P_{1k} \right)^{-1} \Gamma_k P_{2k}^\top H_k, \end{aligned} \tag{2.7a}$$

$$\begin{aligned} (I_{n_1} - H_k G_k)^{-1} &= I_{n_1} + Q_{1k} \left(I_{m_k} - \Sigma_k Q_{2k}^\top G_k Q_{1k} \right)^{-1} \Sigma_k Q_{2k}^\top G_k \\ &= I_{n_1} + H_k P_{1k}\Gamma_k \left(I_{l_k} - P_{2k}^\top H_k P_{1k}\Gamma_k \right)^{-1} P_{2k}^\top. \end{aligned} \tag{2.7b}$$

Denote the direct sum of square matrices by $\oplus$. From (1.5) and (2.7), we can choose the matrices in (2.5) and (2.6) recursively as

$$Q_{1,k+1} = [Q_{1k}, F_k Q_{1k}], \qquad Q_{2,k+1} = [Q_{2k}, E_k^\top Q_{2k}], \tag{2.8}$$

$$P_{1,k+1} = [P_{1k}, E_k P_{1k}], \qquad P_{2,k+1} = [P_{2k}, F_k^\top P_{2k}]; \tag{2.9}$$

$$\begin{aligned} \Sigma_{k+1} &= \Sigma_k \oplus \left[\Sigma_k + (I_{m_k} - \Sigma_k Q_{2k}^\top G_k Q_{1k})^{-1} \Sigma_k Q_{2k}^\top G_k Q_{1k}\Sigma_k \right] \\ &= \Sigma_k \oplus \left[\Sigma_k + \Sigma_k Q_{2k}^\top P_{1k}\Gamma_k (I_{l_k} - P_{2k}^\top H_k P_{1k}\Gamma_k)^{-1} P_{2k}^\top Q_{1k}\Sigma_k \right] \\ &\equiv \Sigma_k \oplus \breve{\Sigma}_k, \end{aligned} \tag{2.10}$$

$$\begin{aligned} \Gamma_{k+1} &= \Gamma_k \oplus \left[\Gamma_k + \Gamma_k P_{2k}^\top Q_{1k}\Sigma_k (I_{m_k} - Q_{2k}^\top G_k Q_{1k}\Sigma_k)^{-1} Q_{2k}^\top P_{1k}\Gamma_k \right] \\ &= \Gamma_k \oplus \left[\Gamma_k + (I_{l_k} - \Gamma_k P_{2k}^\top H_k P_{1k})^{-1} \Gamma_k P_{2k}^\top H_k P_{1k}\Gamma_k \right] \\ &\equiv \Gamma_k \oplus \breve{\Gamma}_k; \end{aligned} \tag{2.11}$$

$$\begin{aligned} F_{1,k+1} &= F_k H_k P_{1k}\Gamma_k (I_{l_k} - P_{2k}^\top H_k P_{1k}\Gamma_k)^{-1} \\ &= F_k Q_{1k}(I_{m_k} - \Sigma_k Q_{2k}^\top G_k Q_{1k})^{-1}\Sigma_k Q_{2k}^\top P_{1k}\Gamma_k, \end{aligned} \tag{2.12}$$

$$F_{2,k+1} = F_k^\top P_{2k}; \tag{2.13}$$

and

$$\begin{aligned} E_{1,k+1} &= E_k G_k Q_{1k}\Sigma_k (I_{m_k} - Q_{2k}^\top G_k Q_{1k}\Sigma_k)^{-1} \\ &= E_k P_{1k}(I_{l_k} - \Gamma_k P_{2k}^\top H_k P_{1k})^{-1}\Gamma_k P_{2k}^\top Q_{1k}\Sigma_k, \end{aligned} \tag{2.14}$$

$$E_{2,k+1} = E_k^\top Q_{2k}. \tag{2.15}$$

Ultimately from (2.6), (2.8), and (2.9), we see that the SDA is dominated by the computation of products like $E_k u$, $E_k^\top u$, $F_k v$, $F_k^\top v$ for arbitrary vectors u and v. By applying (2.6) recursively, these products can be computed using (2.3) and (2.4) in $O(n)$ complexity and memory requirement, because of our assumptions on A, B, C, and D.

The SDA for large-scale NAREs is a competition between the convergence of the doubling iteration and the exponential growth in the dimensions of Q_{ik} and P_{ik}. From (2.5), (2.8), and (2.9), we have

$$\operatorname{rank}(H_k) \le \operatorname{rank}(Q_{ik}) \le 2^k m, \quad \operatorname{rank}(G_k) \le \operatorname{rank}(P_{ik}) \le 2^k l. \tag{2.16}$$

Note that $2^k m$ and $2^k l$ are the numbers of columns in Q_{ik} and P_{ik}, respectively. The success of the SDA depends on the trade-off between the accuracy of the approximate solution and its CPU time and memory requirements, controlled by the compression and truncation of Q_{ik} and P_{ik} in section 2.2. With the truncation and compression process, $\operatorname{rank}(Q_{ik})$ and $\operatorname{rank}(P_{ik})$ will be much reduced even with high accuracy for the approximate solutions H_k and G_k.

From the convergence results in Theorem 1.1, as well as (2.8), (2.9), and (2.16), the fact that X and Y are numerically low ranked can be considered constructively. We next define what we mean by being numerically low ranked of X (and similarly for Y).

Definition 2.1. *Let $X \in \mathbb{C}^{n\times n}$.*

(i) *For a given numerical rank tolerance $\tau > 0$, the numerical rank of X with respect to τ, denoted by $\operatorname{rank}_\tau X$, is defined as the lowest rank of $\widehat{X} \in \mathbb{C}^{n\times n}$ such that $\|\widehat{X} - X\| \leqslant \tau$.*

(ii) *X is said to be numerically low ranked with respect to the numerical rank tolerance $\tau > 0$ if $\operatorname{rank}_\tau(X) \ll n$.*

We first give a useful lemma which is simple but has not appeared in the literature.

Lemma 2.1. *For any $A,\ B \in \mathbb{R}^{n\times r}$, if $0 \le A \le B$, then $\|A\| \le \|B\|$.*

Proof. Since $0 \le A \le B$, we have $0 \le A^\top \le B^\top$ and then $0 \le A^\top A \le B^\top B$. From the Perron–Fronbenius theorem, we have $\|A\| = \sqrt{\rho(A^\top A)} \le \sqrt{\rho(B^\top B)} = \|B\|$. □

2.2. Truncation and compression of Q_{ik} and P_{ik}. The truncation and compression process described in this section is necessary when the convergence of the SDA is slow in comparison with the exponential growth in the dimensions of the iterates G_k and H_k. In this situation, the numerical rank of X will be high and we obviously cannot achieve high accuracy in the approximation H_k of X by any method. We then have to compromise its accuracy for the sake of less memory and

CPU-time consumption. We should then either choose larger tolerances for the truncation and compression process (ε_k below), to control the growth in the iterates and adjust them until the accuracy of the approximate solution is acceptable, or simply abandon the truncation and compression process and accept whatever approximate solution is obtained within reasonable computing constraints.

We now propose a large-scale SDA with truncation error ε (SDA_ls_ε). For a given sequence of tolerances $\varepsilon = \{\varepsilon_k\}_{k=0}^{\bar{k}}$ we first compute truncated initial matrices for the SDA_ls_ε. From (2.2) we compute the QR decompositions

$$Q_{10} = \widetilde{Q}_{10} R_{1q}, \quad Q_{20} = \widetilde{Q}_{20} R_{2q}, \quad P_{10} = \widetilde{P}_{10} R_{1p}, \quad P_{20} = \widetilde{P}_{20} R_{2p},$$

where $\widetilde{Q}_{10}$, $\widetilde{Q}_{20}$, $\widetilde{P}_{10}$, and $\widetilde{P}_{20}$ are orthogonal and R_{1q}, R_{2q}, R_{1p}, and R_{2p} are upper triangular. Then we compute the SVD decompositions

$$\begin{aligned} R_{1q}\Sigma_0 R_{2q}^\top &= [U_{10}^\tau, U_{10}^\varepsilon](\Sigma_0^\tau \oplus \Sigma_0^\varepsilon)[U_{20}^\tau, U_{20}^\varepsilon]^\top, \quad && \|\Sigma_0^\varepsilon\| < \varepsilon_0, \\ R_{1p}\Gamma_0 R_{2p}^\top &= [V_{10}^\tau, V_{10}^\varepsilon](\Gamma_0^\tau \oplus \Gamma_0^\varepsilon)[V_{20}^\tau, V_{20}^\varepsilon]^\top, \quad && \|\Gamma_0^\varepsilon\| < \varepsilon_0, \end{aligned}$$

where $[U_{i0}^\tau, U_{i0}^\varepsilon]$ and $[V_{i0}^\tau, V_{i0}^\varepsilon]$ $(i = 1, 2)$ are orthogonal, $\Sigma_0^\tau \oplus \Sigma_0^\varepsilon$ and $\Gamma_0^\tau \oplus \Gamma_0^\varepsilon$ are nonnegative diagonal with $\Sigma_0^\tau \in \mathbb{R}^{m_0 \times m_0}$ and $\Gamma_0^\tau \in \mathbb{R}^{l_0 \times l_0}$. By setting

$$Q_{10}^\tau = \widetilde{Q}_{10} U_{10}^\tau, \quad Q_{20}^\tau = \widetilde{Q}_{20} U_{20}^\tau, \quad P_{10}^\tau = \widetilde{P}_{10} V_{10}^\tau, \quad P_{20}^\tau = \widetilde{P}_{20} V_{20}^\tau, \tag{2.17}$$

we have the (truncated) initial matrices

$$F_0^\tau = F_0, \quad E_0^\tau = E_0, \quad H_0^\tau = Q_{10}^\tau \Sigma_0^\tau Q_{20}^{\tau\,\top}, \quad G_0^\tau = P_{10}^\tau \Gamma_0^\tau P_{20}^{\tau\,\top} \tag{2.18}$$

for the SDA_ls_ε. Let

$$\Delta H_0 \equiv H_0 - H_0^\tau = \widetilde{Q}_{10} U_{10}^\varepsilon \Sigma_0^\varepsilon U_{20}^{\varepsilon\,\top} \widetilde{Q}_{20}^\top, \quad \Delta G_0 \equiv G_0 - G_0^\tau = \widetilde{P}_{10} V_{10}^\varepsilon \Gamma_0^\varepsilon V_{20}^{\varepsilon\,\top} \widetilde{P}_{20}^\top. \tag{2.19}$$

The truncation errors of the initial matrices can be estimated by

$$\|\Delta H_0\| = \|\Sigma_0^\varepsilon\| < \varepsilon_0, \quad \|\Delta G_0\| = \|\Gamma_0^\varepsilon\| < \varepsilon_0. \tag{2.20}$$

We repeat this process and suppose it holds at the k step that

$$\begin{aligned} H_k^\tau &= Q_{1k}^\tau \Sigma_k^\tau Q_{2k}^{\tau\,\top}, & G_k^\tau &= P_{1k}^\tau \Gamma_k^\tau P_{2k}^{\tau\,\top}, \\ E_k^\tau &= E_{k-1}^{\tau^2} + E_{1k}^\tau E_{2k}^{\tau\,\top}, & F_k^\tau &= F_{k-1}^{\tau^2} + F_{1k}^\tau F_{2k}^{\tau\,\top}, \end{aligned} \tag{2.21}$$

where Q_{ik}^τ and P_{ik}^τ are orthogonal with widths being m_k and l_k $(i = 1, 2)$, respectively.

To estimate the $(k+1)$th truncation error, from (2.10)–(2.15) as well as (2.21), we compute

$$\begin{aligned} \breve{\Sigma}_k^\tau &= \Sigma_k^\tau + \Sigma_k^\tau Q_{2k}^{\tau\,\top} P_{1k}^\tau \Gamma_k^\tau (I_{l_k} - P_{2k}^{\tau\,\top} H_k^\tau P_{1k}^\tau \Gamma_k^\tau)^{-1} P_{2k}^{\tau\,\top} Q_{1k}^\tau \Sigma_k^\tau, \\ \breve{\Gamma}_k^\tau &= \Gamma_k^\tau + (I_{l_k} - \Gamma_k^\tau P_{2k}^{\tau\,\top} H_k^\tau P_{1k}^\tau)^{-1} \Gamma_k^\tau P_{2k}^{\tau\,\top} H_k^\tau P_{1k}^\tau \Gamma_k^\tau \end{aligned} \tag{2.22}$$

and

$$\begin{aligned} F_{1,k+1}^\tau &= F_k^\tau H_k^\tau P_{1k}^\tau \Gamma_k^\tau \left(I_{l_k} - P_{2k}^{\tau\top} H_k^\tau P_{1k}^\tau \Gamma_k^\tau\right)^{-1}, & F_{2,k+1}^\tau &= F_k^{\tau\,\top} P_{2k}^\tau, \\ E_{1,k+1}^\tau &= E_k^\tau P_{1k}^\tau \left(I_{l_k} - \Gamma_k^\tau P_{2k}^{\tau\,\top} H_k^\tau P_{1k}^\tau\right)^{-1} \Gamma_k^\tau P_{2k}^{\tau\,\top} Q_{1k}^\tau \Sigma_k^\tau, & E_{2,k+1}^\tau &= E_k^{\tau\,\top} Q_{2k}^\tau. \end{aligned} \tag{2.23}$$

From the QR decompositions

$$[Q_{1k}^{\tau}, F_k^{\tau} Q_{1k}^{\tau}] = [Q_{1k}^{\tau}, \widetilde{Q}_{1k}] \begin{bmatrix} I & S_{1q} \\ 0 & R_{1q} \end{bmatrix}_k, \quad [Q_{2k}^{\tau}, E_k^{\tau\top} Q_{2k}^{\tau}] = [Q_{2k}^{\tau}, \widetilde{Q}_{2k}] \begin{bmatrix} I & S_{2q} \\ 0 & R_{2q} \end{bmatrix}_k,$$
$$[P_{1k}^{\tau}, E_k^{\tau} P_{1k}^{\tau}] = [P_{1k}^{\tau}, \widetilde{P}_{1k}] \begin{bmatrix} I & S_{1p} \\ 0 & R_{1p} \end{bmatrix}_k, \quad [P_{2k}^{\tau}, F_k^{\tau\top} P_{2k}^{\tau}] = [P_{2k}^{\tau}, \widetilde{P}_{2k}] \begin{bmatrix} I & S_{2p} \\ 0 & R_{2p} \end{bmatrix}_k, \tag{2.24}$$

we set

$$\widehat{\Sigma}_{k+1} \equiv \begin{bmatrix} I & S_{1q} \\ 0 & R_{1q} \end{bmatrix}_k \begin{bmatrix} \Sigma_k^{\tau} & 0 \\ 0 & \breve{\Sigma}_k^{\tau} \end{bmatrix} \begin{bmatrix} I & S_{2q} \\ 0 & R_{2q} \end{bmatrix}_k^{\top}, \quad \widehat{\Gamma}_{k+1} \equiv \begin{bmatrix} I & S_{1p} \\ 0 & R_{1p} \end{bmatrix}_k \begin{bmatrix} \Gamma_k^{\tau} & 0 \\ 0 & \breve{\Gamma}_k^{\tau} \end{bmatrix} \begin{bmatrix} I & S_{2p} \\ 0 & R_{2p} \end{bmatrix}_k^{\top}. \tag{2.25}$$

We next compute the SVDs

$$\widehat{\Sigma}_{k+1} = \begin{bmatrix} U_{1,k+1}^{\tau} & U_{1,k+1}^{\varepsilon} \end{bmatrix} \begin{bmatrix} \Sigma_{k+1}^{\tau} & 0 \\ 0 & \Sigma_{k+1}^{\varepsilon} \end{bmatrix} \begin{bmatrix} U_{2,k+1}^{\tau} & U_{2,k+1}^{\varepsilon} \end{bmatrix}^{\top},$$
$$\widehat{\Gamma}_{k+1} = \begin{bmatrix} V_{1,k+1}^{\tau} & V_{1,k+1}^{\varepsilon} \end{bmatrix} \begin{bmatrix} \Gamma_{k+1}^{\tau} & 0 \\ 0 & \Gamma_{k+1}^{\varepsilon} \end{bmatrix} \begin{bmatrix} V_{2,k+1}^{\tau} & V_{2,k+1}^{\varepsilon} \end{bmatrix}^{\top} \tag{2.26}$$

with $\|\Sigma_{k+1}^{\varepsilon}\| < \varepsilon_{k+1}$ and $\|\Gamma_{k+1}^{\varepsilon}\| < \varepsilon_{k+1}$. To truncate, we compute

$$Q_{1,k+1}^{\tau} \equiv [Q_{1k}^{\tau}, \widetilde{Q}_{1k}] U_{1,k+1}^{\tau} \in \mathbb{R}^{n_1 \times m_{k+1}}, \quad Q_{2,k+1}^{\tau} \equiv [Q_{2k}^{\tau}, \widetilde{Q}_{2k}] U_{2,k+1}^{\tau} \in \mathbb{R}^{n_2 \times m_{k+1}},$$
$$P_{1,k+1}^{\tau} \equiv [P_{1k}^{\tau}, \widetilde{P}_{1k}] V_{1,k+1}^{\tau} \in \mathbb{R}^{n_2 \times l_{k+1}}, \quad P_{2,k+1}^{\tau} \equiv [P_{2k}^{\tau}, \widetilde{P}_{2k}] V_{2,k+1}^{\tau} \in \mathbb{R}^{n_1 \times l_{k+1}}. \tag{2.27}$$

Let

$$\widehat{H}_{k+1} = H_k^{\tau} + F_k^{\tau}(I - H_k^{\tau} G_k^{\tau})^{-1} H_k^{\tau} E_k^{\tau}, \quad \widehat{G}_{k+1} = G_k^{\tau} + E_k^{\tau}(I - G_k^{\tau} H_k^{\tau})^{-1} G_k^{\tau} F_k^{\tau}. \tag{2.28}$$

We define the local truncation errors of H_{k+1}^{τ} and G_{k+1}^{τ} as

$$\delta H_{k+1} \equiv \widehat{H}_{k+1} - H_{k+1}^{\tau}, \quad \delta G_{k+1} \equiv \widehat{G}_{k+1} - G_{k+1}^{\tau}. \tag{2.29}$$

From (2.26), we see that

$$\|\delta H_{k+1}\| = \left\| [Q_{1k}^{\tau}, \widetilde{Q}_{1k}] U_{1,k+1}^{\varepsilon} \Sigma_{k+1}^{\varepsilon} U_{2,k+1}^{\varepsilon\top} [Q_{2k}^{\tau}, \widetilde{Q}_{2k}]^{\top} \right\| = \|\Sigma_{k+1}^{\varepsilon}\| < \varepsilon_{k+1}, \tag{2.30}$$
$$\|\delta G_{k+1}\| = \left\| [P_{1k}^{\tau}, \widetilde{P}_{1k}] V_{1,k+1}^{\varepsilon} \Gamma_{k+1}^{\varepsilon} V_{2,k+1}^{\varepsilon\top} [P_{2k}^{\tau}, \widetilde{P}_{2k}]^{\top} \right\| = \|\Gamma_{k+1}^{\varepsilon}\| < \varepsilon_{k+1}. \tag{2.31}$$

Moreover, we define the global truncation errors of H_{k+1}^{τ} and G_{k+1}^{τ} by

$$\Delta H_{k+1} \equiv H_{k+1} - H_{k+1}^{\tau}, \quad \Delta G_{k+1} \equiv G_{k+1} - G_{k+1}^{\tau}, \tag{2.32}$$

which will be estimated in section 3.

Algorithm SDA_ls_ε for solving large-scale NAREs realizes the iterations in (1.5) with initial matrices in (2.18), and the help of (2.3), (2.4), (2.21)–(2.27).

Algorithm 1. SDA_ls_ε.

```
Input:   $A \in \mathbb{R}^{n_1\times n_1}$, $B \in \mathbb{R}^{n_1\times n_2}$, $C \in \mathbb{R}^{n_2\times n_1}$, $D \in \mathbb{R}^{n_2\times n_2}$ with $B = B_1B_2^\top$ and
         $C = C_1C_2^\top$ being full low ranked as in (2.1); suitable shift $\gamma$ as in (1.3);
         truncation tolerances $\varepsilon = \{\varepsilon_k\}_{k=0}^{\bar{k}}$ and convergence tolerance $\varepsilon_c > 0$.
Output:  $H_{\bar{k}}^\tau = Q_{1\bar{k}}^\tau \Sigma_{\bar{k}}^\tau Q_{2\bar{k}}^{\tau\top}$ and $G_{\bar{k}}^\tau = P_{1\bar{k}}^\tau \Gamma_{\bar{k}}^\tau P_{2\bar{k}}^{\tau\top}$ with $Q_{i\bar{k}}^\tau \in \mathbb{R}^{n_i\times m_{\bar{k}}}$, $P_{j_i,\bar{k}}^\tau \in \mathbb{R}^{n_i\times l_{\bar{k}}}$
         orthogonal ($i = 1, 2; j_1 = 2, j_2 = 1$), approximating the solutions $X$ and $Y$
         to the large-scale NARE (1.1) and its dual equation (1.6), respectively.
Initial matrices:
         Set $k = 0$;
         Compute $Q_{i0}, P_{i0}$ $(i = 1, 2)$ in (2.2);
         Compute $Q_{i0}^\tau, P_{i0}^\tau$ $(i = 1, 2)$ in (2.17) with truncation tolerance $\varepsilon_0$;
         Do until convergence:
           Compute $\breve{\Sigma}_k^\tau$, $\breve{\Gamma}_k^\tau$ as in (2.22) and $F_{i,k+1}^\tau$, $E_{i,k+1}^\tau$ $(i = 1, 2)$ as in (2.23);
           Orthogonalize $F_k^\tau Q_{1k}^\tau, E_k^{\tau\top} Q_{2k}^\tau, E_k^\tau P_{1k}^\tau$, and $F_k^{\tau\top} P_{2k}^\tau$ as in (2.24);
           Compute $\widehat{\Sigma}_{k+1}$ and $\widehat{\Gamma}_{k+1}$ as in (2.25), and their SVDs as in (2.26);
           Compute $Q_{i,k+1}^\tau$, $P_{i,k+1}^\tau$ $(i = 1, 2)$ using the tolerance $\varepsilon_{k+1}$ as in (2.27);
           Compute $k \leftarrow k + 1$, $d_k = \max\{\|dH_k^\tau\|, \|dG_k^\tau\|\}$ and $r_k = \|\mathcal{R}(H_k^\tau)\|$;
           (An economic way for computing $\|dH_k^\tau\|$, $\|dG_k^\tau\|$, and $\|\mathcal{R}(H_k^\tau)\|$ can be
           found in (4.1), (4.2), and (4.3) in section 4.1.)
           If $d_k < \varepsilon_c$, Set $\bar{k} = k$; Stop; End If;
         End Do
```

2.3. SDA and Krylov subspaces. There is an interesting relationship between the SDA and Krylov subspaces. Define the Krylov subspaces

$$\mathcal{K}_k(A, V) \equiv \begin{cases} \operatorname{span}\{V\} & (k = 0), \\ \operatorname{span}\{V, AV, A^2V, \ldots, A^{2^k-1}V\} & (k > 0). \end{cases}$$

From (1.4), (2.2)–(2.4), and (2.8), we see that

$$\begin{aligned} Q_{10} &= 2\gamma W_\gamma^{-1} B_1 \subseteq \mathcal{K}_0(A_\gamma^{-1},\, A_\gamma^{-1}B_1), \qquad & Q_{20} &= D_\gamma^{-\top} B_2 \subseteq \mathcal{K}_0(D_\gamma^{-\top},\, D_\gamma^{-\top}B_2), \\ Q_{11} &= [Q_{10}, F_0 Q_{10}] \subseteq \mathcal{K}_1(A_\gamma^{-1},\, A_\gamma^{-1}B_1), \qquad & Q_{21} &= [Q_{20}, E_0^\top Q_{20}] \subseteq \mathcal{K}_1(D_\gamma^{-\top},\, D_\gamma^{-\top}B_2). \end{aligned}$$

(We have abused notation, with $V \subseteq \mathcal{K}_k(A, B)$ meaning $\operatorname{span}\{V\} \subseteq \mathcal{K}_k(A, B)$.) Similarly, it is easy to show that

$$\begin{aligned} Q_{1k} &\subseteq \mathcal{K}_k(A_\gamma^{-1},\, A_\gamma^{-1}B_1), \quad & Q_{2k} &\subseteq \mathcal{K}_k(D_\gamma^{-\top},\, D_\gamma^{-\top}B_2), \\ P_{1k} &\subseteq \mathcal{K}_k(D_\gamma^{-1},\, D_\gamma^{-1}C_1), \quad & P_{2k} &\subseteq \mathcal{K}_k(A_\gamma^{-\top},\, A_\gamma^{-\top}C_2). \end{aligned}$$

In other words, the general SDA is closely related to approximating the solutions X and Y using Krylov subspaces, with additional components diminishing quadratically. However, for problems of moderate size n, Q_{ik} and P_{ik} become full-ranked after a few iterations.

The link between the SDA and the Krylov subspaces defined above is important in explaining the fast convergence of the SDA. We used to believe the convergence of the SDA came from the following inequalities:

$$\begin{aligned} \|H_k - H_{k-1}\| &\le \|F_{k-1}\| \|(I_{n_1} - H_{k-1}G_{k-1})^{-1} H_{k-1}\| \|E_{k-1}\|, \\ \|G_k - G_{k-1}\| &\le \|E_{k-1}\| \|(I_{n_2} - G_{k-1}H_{k-1})^{-1} G_{k-1}\| \|F_{k-1}\|, \end{aligned}$$

and the fact that $\|E_{k-1}\|, \|F_{k-1}\| \to 0$ quadratically, as $k \to \infty$. This is consistent with numerical results from examples associated with M in (1.2) which is barely a nonsingular M-matrix, where the corresponding $E_k, F_k \to 0$ slowly but the overall convergence for H_k and G_k is much faster.

3. Truncation error estimates.

In this section, we shall estimate the global truncation errors defined in (2.32). For simplicity, we derive only the first order error bounds.

From Theorem 1.1, we have $0 \le H_k \le X,\ 0 \le G_k \le Y$, and

$$\begin{aligned} 0 \le (I - G_k H_k)^{-1} &= I + G_k H_k + (G_k H_k)^2 + \cdots \\ &\le I + YX + (YX)^2 + \cdots = (I - YX)^{-1} \end{aligned}$$

and $0 \le F_k = (I_{n_1} - H_k Y) S_\gamma^{2^k} \le S_\gamma^{2^k}$. By Lemma 2.1, we have

$$\|H_k\| \le \|X\|,\ \|G_k\| \le \|Y\|, \tag{3.1}$$

and

$$\|(I - G_k H_k)^{-1}\| \le \|(I - YX)^{-1}\| \equiv \beta_1, \quad \|F_k\| \le \|S_\gamma^{2^k}\| \to 0. \tag{3.2}$$

Similarly, from Theorem 1.1 and Lemma 2.1 we also have

$$\|(I - H_k G_k)^{-1}\| \le \|(I - XY)^{-1}\| \equiv \beta_2, \quad \|E_k\| \le \|R_\gamma^{2^k}\| \to 0. \tag{3.3}$$

Denote

$$\rho_k = \max\{\|R_\gamma^{2^k}\|, \|S_\gamma^{2^k}\|\}, \quad \alpha = \max\{\|X\|, \|Y\|\}, \quad \beta = \max\{\beta_1, \beta_2\}. \tag{3.4}$$

In the following we abuse the notation "=" and "≡" by ignoring the higher order terms. Suppose that $\rho(H_k^\tau G_k^\tau) < 1$, $\|\Delta H_k\|$ and $\|\Delta G_k\|$ are sufficiently small. From (2.32) we have the first order approximation of $(I - H_k^\tau G_k^\tau)^{-1}$:

$$\begin{aligned} (I - H_k^\tau G_k^\tau)^{-1} &= [I - (H_k - \Delta H_k)(G_k - \Delta G_k)]^{-1} \\ &= [I - H_k G_k + \Delta H_k G_k + H_k \Delta G_k - \Delta H_k \Delta G_k]^{-1} \\ &= (I - H_k G_k)^{-1} - (I - H_k G_k)^{-1}(\Delta H_k G_k + H_k \Delta G_k)(I - H_k G_k)^{-1}. \end{aligned}$$

From (2.19) and (2.20), we have $H_0^\tau = H_0 - \Delta H_0,\ G_0^\tau = G_0 - \Delta G_0$ with $\|\Delta H_0\|$, $\|\Delta G_0\| < \varepsilon_0$. Since $E_0^\tau = E_0$ and $F_0^\tau = F_0$, (2.28) implies

$$\begin{aligned} \widehat{H}_1 &= H_0^\tau + F_0^\tau (I - H_0^\tau G_0^\tau)^{-1} H_0^\tau E_0^\tau \\ &= H_0 - \Delta H_0 \\ &\quad + F_0 \left[(I - H_0 G_0)^{-1} - (I - H_0 G_0)^{-1}(\Delta H_0 G_0 + H_0 \Delta G_0)(I - H_0 G_0)^{-1}\right] \\ &\qquad \times (H_0 - \Delta H_0) E_0 \\ &= H_0 + F_0 (I - H_0 G_0)^{-1} H_0 E_0 - \Delta H_0 - F_0 (I - H_0 G_0)^{-1} \Delta H_0 E_0 \\ &\quad - F_0 (I - H_0 G_0)^{-1}(\Delta H_0 G_0 + H_0 \Delta G_0)(I - H_0 G_0)^{-1} H_0 E_0 \\ &\equiv H_1 - \widehat{\delta} H_1, \end{aligned} \tag{3.5}$$

where $\widehat{\delta} H_1$ is the first order truncation error given by

$$\begin{aligned} \widehat{\delta} H_1 &= \Delta H_0 + F_0 (I - H_0 G_0)^{-1} \Delta H_0 E_0 \\ &\quad + F_0 (I - H_0 G_0)^{-1}(\Delta H_0 G_0 + H_0 \Delta G_0)(I - H_0 G_0)^{-1} H_0 E_0. \end{aligned}$$

From (2.29), (2.32), and (3.5), it follows that $\Delta H_1 = H_1 - H_1^\tau = \widehat{\delta} H_1 + \delta H_1$. By (2.30) and (3.1)–(3.4), we have

$$\begin{aligned}
\|\Delta H_1\| &\le \|\delta H_1\| + \|\widehat{\delta} H_1\| \\
&\le \varepsilon_1 + \|\Delta H_0\| + \|F_0\|\|E_0\|\|(I - H_0G_0)^{-1}\|\|\Delta H_0\| \\
&\quad + \|F_0\|\|E_0\|\|H_0\|\|(I - H_0G_0)^{-1}\|^2(\|\Delta H_0\|\|G_0\| + \|H_0\|\|\Delta G_0\|) \\
&\le \varepsilon_1 + (1 + \rho_0^2\beta + \rho_0^2\alpha^2\beta^2)\|\Delta H_0\| + \rho_0^2\alpha^2\beta^2\|\Delta G_0\|.
\end{aligned}$$

Similarly, we have

$$\|\Delta G_1\| \le \varepsilon_1 + (1 + \rho_0^2\beta + \rho_0^2\alpha^2\beta^2)\|\Delta G_0\| + \rho_0^2\alpha^2\beta^2\|\Delta H_0\|.$$

From (2.21), we have

$$\begin{aligned}
F_1^\tau &= F_0^\tau(I - H_0^\tau G_0^\tau)^{-1}F_0^\tau = F_0(I - H_0^\tau G_0^\tau)^{-1}F_0 \\
&= F_0(I - H_0G_0)^{-1}F_0 - F_0(I - H_0G_0)^{-1}(\Delta H_0G_0 + H_0\Delta G_0)(I - H_0G_0)^{-1}F_0 \\
&\equiv F_1 - \Delta F_1,
\end{aligned}$$

where $\Delta F_1 = F_0(I - H_0G_0)^{-1}(\Delta H_0G_0 + H_0\Delta G_0)(I - H_0G_0)^{-1}F_0$ is the first order truncation error and satisfies

$$\begin{aligned}
\|\Delta F_1\| &\le \|F_0\|^2\|(I - H_0G_0)^{-1}\|^2(\|\Delta H_0\|\|G_0\| + \|H_0\|\|\Delta G_0\|) \\
&\le \rho_0^2\alpha\beta^2(\|\Delta H_0\| + \|\Delta G_0\|).
\end{aligned}$$

Similarly, we also have

$$\|\Delta E_1\| \le \rho_0^2\alpha\beta^2(\|\Delta H_0\| + \|\Delta G_0\|).$$

Performing the $(k+1)$th step in the SDA_ls_ε algorithm, we obtain

$$\begin{aligned}
\widehat{H}_{k+1} &= H_k^\tau + F_k^\tau(I - H_k^\tau G_k^\tau)^{-1}H_k^\tau E_k^\tau \\
&= H_k - \Delta H_k + (F_k - \Delta F_k)(I - H_kG_k)^{-1}(H_k - \Delta H_k)(E_k - \Delta E_k) \\
&\quad - (F_k - \Delta F_k)(I - H_kG_k)^{-1}(\Delta H_kG_k + H_k\Delta G_k)(I - H_kG_k)^{-1} \\
&\quad\quad (H_k - \Delta H_k)(E_k - \Delta E_k) \\
&= H_k + F_k(I - H_kG_k)^{-1}H_kE_k - \Delta H_k - F_k(I - H_kG_k)^{-1}(H_k\Delta E_k + \Delta H_kE_k) \\
&\quad - \Delta F_k(I - H_kG_k)^{-1}H_kE_k - F_k(I - H_kG_k)^{-1}(\Delta H_kG_k + H_k\Delta G_k) \\
&\quad\quad \times (I - H_kG_k)^{-1}H_kE_k \\
&\equiv H_{k+1} - \widehat{\delta} H_{k+1},
\end{aligned}$$

where

$$\begin{aligned}
\widehat{\delta} H_{k+1} &= \Delta H_k + F_k(I - H_kG_k)^{-1}(H_k\Delta E_k + \Delta H_kE_k) + \Delta F_k(I - H_kG_k)^{-1}H_kE_k \\
&\quad + F_k(I - H_kG_k)^{-1}(\Delta H_kG_k + H_k\Delta G_k)(I - H_kG_k)^{-1}H_kE_k
\end{aligned}$$

is the first order truncation error. Then (2.30)–(2.32) and (3.1)–(3.4) imply

$$\begin{aligned}
\|\Delta H_{k+1}\| &\le \|\delta H_{k+1}\| + \|\widehat{\delta} H_{k+1}\| \\
&\le \|\delta H_{k+1}\| + \|\Delta H_k\| + \|(I - H_kG_k)^{-1}\|\|F_k\|(\|H_k\|\|\Delta E_k\| + \|E_k\|\|\Delta H_k\|) \\
&\quad + \|(I - H_kG_k)^{-1}\|\|H_k\|\|E_k\|\|\Delta F_k\| \\
&\quad + \|F_k\|\|E_k\|\|H_k\|\|(I - H_kG_k)^{-1}\|^2(\|\Delta H_k\|\|G_k\| + \|H_k\|\|\Delta G_k\|) \\
&\le \varepsilon_{k+1} + (1 + \rho_k^2\beta + \rho_k^2\alpha^2\beta^2)\|\Delta H_k\| \\
&\quad + \rho_k^2\alpha^2\beta^2\|\Delta G_k\| + \rho_k\alpha\beta(\|\Delta E_k\| + \|\Delta F_k\|). \qquad (3.6)
\end{aligned}$$

Similarly, we also have

$$\|\Delta G_{k+1}\| \le \varepsilon_{k+1}+(1+\rho_k^2\beta+\rho_k^2\alpha^2\beta^2)\|\Delta G_k\|+\rho_k^2\alpha^2\beta^2\|\Delta H_k\|+\rho_k\alpha\beta(\|\Delta F_k\|+\|\Delta E_k\|). \tag{3.7}$$

From (2.21), it holds that

$$\begin{aligned} F_{k+1}^\tau &= F_k^\tau(I-H_k^\tau G_k^\tau)^{-1}F_k^\tau = (F_k-\Delta F_k)(I-H_k^\tau G_k^\tau)^{-1}(F_k-\Delta F_k) \\ &= F_k(I-H_kG_k)^{-1}F_k - F_k(I-H_kG_k)^{-1}\Delta F_k - \Delta F_k(I-H_kG_k)^{-1}F_k \\ &\quad - F_k(I-H_kG_k)^{-1}(\Delta H_kG_k+H_k\Delta G_k)(I-H_kG_k)^{-1}F_k \\ &\equiv F_{k+1}-\Delta F_{k+1}, \end{aligned}$$

where

$$\begin{aligned} \Delta F_{k+1} =& F_k(I-H_kG_k)^{-1}\Delta F_k + \Delta F_k(I-H_kG_k)^{-1}F_k \\ &+ F_k(I-H_kG_k)^{-1}(\Delta H_kG_k+H_k\Delta G_k)(I-H_kG_k)^{-1}F_k \end{aligned}$$

is the first order truncation error and satisfies

$$\begin{aligned} \|\Delta F_{k+1}\| &\le 2\|F_k\|\|(I-H_kG_k)^{-1}\|\|\Delta F_k\| \\ &\quad + \|F_k\|^2\|(I-H_kG_k)^{-1}\|^2(\|\Delta H_k\|\|G_k\|+\|H_k\|\|\Delta G_k\|) \\ &\le 2\rho_k\beta\|\Delta F_k\| + \rho_k^2\alpha\beta^2(\|\Delta H_k\|+\|\Delta G_k\|). \end{aligned} \tag{3.8}$$

Similarly, we also have

$$\|\Delta E_{k+1}\| \le 2\rho_k\beta\|\Delta E_k\| + \rho_k^2\alpha\beta^2(\|\Delta H_k\|+\|\Delta G_k\|). \tag{3.9}$$

Assembling (3.6)–(3.9) in matrix form, we have

$$\begin{aligned} &\begin{bmatrix} \|\Delta H_{k+1}\| \\ \|\Delta G_{k+1}\| \\ \|\Delta F_{k+1}\| \\ \|\Delta E_{k+1}\| \end{bmatrix} \le \begin{bmatrix} 1+\rho_k^2\beta+\rho_k^2\alpha^2\beta^2 & \rho_k^2\alpha^2\beta^2 & \rho_k\alpha\beta & \rho_k\alpha\beta \\ \rho_k^2\alpha^2\beta^2 & 1+\rho_k^2\beta+\rho_k^2\alpha^2\beta^2 & \rho_k\alpha\beta & \rho_k\alpha\beta \\ \rho_k^2\alpha\beta^2 & \rho_k^2\alpha\beta^2 & 2\rho_k\beta & 0 \\ \rho_k^2\alpha\beta^2 & \rho_k^2\alpha\beta^2 & 0 & 2\rho_k\beta \end{bmatrix} \begin{bmatrix} \|\Delta H_k\| \\ \|\Delta G_k\| \\ \|\Delta F_k\| \\ \|\Delta E_k\| \end{bmatrix} + \begin{bmatrix} \varepsilon_{k+1} \\ \varepsilon_{k+1} \\ 0 \\ 0 \end{bmatrix} \\ &\equiv \Psi_k[\|\Delta H_k\|, \|\Delta G_k\|, \|\Delta F_k\|, \|\Delta E_k\|]^\top + [\varepsilon_{k+1}, \varepsilon_{k+1}, 0, 0]^\top. \end{aligned} \tag{3.10}$$

Substituting Ψ_k in (3.10) recursively, the error bound can be estimated by

$$\begin{bmatrix} \|\Delta H_{k+1}\| \\ \|\Delta G_{k+1}\| \\ \|\Delta F_{k+1}\| \\ \|\Delta E_{k+1}\| \end{bmatrix} \le \sum_{i=1}^{k+1}\left(\prod_{j=0}^{i-1}\Psi_{k-j}\begin{bmatrix} \varepsilon_{k-i+1} \\ \varepsilon_{k-i+1} \\ 0 \\ 0 \end{bmatrix}\right) + \begin{bmatrix} \varepsilon_{k+1} \\ \varepsilon_{k+1} \\ 0 \\ 0 \end{bmatrix}. \tag{3.11}$$

In the following theorem we claim that the first order forward error bounds of H_k^τ, G_k^τ and the first order truncation errors of F_k^τ and E_k^τ, only depend on ρ_k and the tolerance $\varepsilon = \{\varepsilon_k\}_{k=0}^{\bar{k}}$.

THEOREM 3.1. *Let X and Y be the minimal nonnegative solutions of NARE* (1.1) *and its dual equation* (1.6)*, respectively. For given tolerances $\varepsilon = \{\varepsilon_k\}_{k=0}^{\bar{k}}$, suppose $\{H_k^\tau, G_k^\tau, F_k^\tau, E_k^\tau\}_{k=0}^{\bar{k}}$ is the sequence generated by the SDA_ls_ε satisfying $\rho(H_k^\tau G_k^\tau) < 1$ for all k. Then we have*

$$\|H_k^\tau - X\|, \|G_k^\tau - Y\| \le \varepsilon_k + \frac{1}{2}\sum_{i=1}^{k}\left[1+\prod_{j=1}^{i}(1+\eta_{k-j})\right]\varepsilon_{k-i} + \rho_k^2\alpha, \tag{3.12}$$

and

$$\|F_k^\tau - F_k\|, \|E_k^\tau - E_k\| \leq \frac{1}{2}\sum_{i=1}^{k}\left[1 + \prod_{j=1}^{i}(1+\eta_{k-j})\right]\varepsilon_{k-i} \tag{3.13}$$

for $k = 0, 1, \cdots \bar{k}$, where ρ_k is given by (3.4) and η_k is defined by

$$\eta_k = 4\max\{\rho_k^2\beta + \rho_k^2\alpha^2\beta^2,\ \rho_k\alpha\beta,\ \rho_k^2\alpha\beta^2,\ 2\rho_k\beta\}. \tag{3.14}$$

Proof. For convenience, we let

$$\widehat{\Psi}_k = \operatorname{diag}(1,1,0,0) + \frac{1}{4}\eta_k ee^\top \equiv J_0 + \eta_k J_1,$$

where $e = [1,1,1,1]^\top$ and η_k is given in (3.14). Then (3.11) can be simplified to

$$\begin{bmatrix}\|\Delta H_{k+1}\|\\ \|\Delta G_{k+1}\|\\ \|\Delta F_{k+1}\|\\ \|\Delta E_{k+1}\|\end{bmatrix} \leq \sum_{i=1}^{k+1}\left(\prod_{j=0}^{i-1}\widehat{\Psi}_{k-j}\begin{bmatrix}\varepsilon_{k-i+1}\\ \varepsilon_{k-i+1}\\ 0\\ 0\end{bmatrix}\right) + \begin{bmatrix}\varepsilon_{k+1}\\ \varepsilon_{k+1}\\ 0\\ 0\end{bmatrix}. \tag{3.15}$$

It is easily seen that

$$J_0^2 = J_0, \quad J_1^2 = J_1, \quad J_0J_1J_0 \leq J_1, \quad J_1J_0J_1 \leq J_1. \tag{3.16}$$

Let $\mathcal{C}(J_0^s, J_1^t)$ denote the product of s's J_0 and t's J_1 in any order. From (3.16), it follows that

$$\mathcal{C}(J_0^s, J_1^t) \leq \begin{cases} J_1 & \text{if } t \geq 1,\\ J_0 & \text{if } t = 0.\end{cases} \tag{3.17}$$

By (3.17), the products in (3.15) can be bounded by

$$\prod_{j=0}^{i-1}\widehat{\Psi}_{k-j} = \widehat{\Psi}_k \cdots \widehat{\Psi}_{k-i+1} \leq J_0 + \sum_{t=1}^{i}\left(\sum_{k\geq r_1>\cdots>r_t\geq k-i+1}\eta_{r_1}\cdots\eta_{r_t}\right)J_1. \tag{3.18}$$

Postmultiplying (3.18) by $[\varepsilon_{k-i+1}, \varepsilon_{k-i+1}, 0, 0]^\top$ and substituting the result into (3.15), we obtain the first order upper bounds

$$\begin{aligned}\|\Delta H_{k+1}\| &\leq \varepsilon_{k+1} + \sum_{i=1}^{k+1}\left[1 + \frac{1}{2}\sum_{t=1}^{i}\left(\sum_{k\geq r_1>\cdots>r_t\geq k-i+1}\eta_{r_1}\cdots\eta_{r_t}\right)\right]\varepsilon_{k-i+1}\\ &= \varepsilon_{k+1} + \frac{1}{2}\sum_{i=1}^{k+1}\left[1 + \prod_{j=0}^{i-1}(1+\eta_{k-j})\right]\varepsilon_{k-i+1}, \end{aligned} \tag{3.19}$$

$$\|\Delta F_{k+1}\| \leq \frac{1}{2}\sum_{i=1}^{k+1}\left[1 + \prod_{j=0}^{i-1}(1+\eta_{k-j})\right]\varepsilon_{k-i+1}, \tag{3.20}$$

which also hold for $\|\Delta G_{k+1}\|$ and $\|\Delta E_{k+1}\|$, respectively. By Theorem 1.1 and (3.4), we have $\|H_k - X\| \leq \rho_k^2\alpha$ and $\|G_k - Y\| \leq \rho_k^2\alpha$. Since $\|H_k^\tau - X\| \leq \|H_k^\tau - H_k\| +$

$\|H_k - X\|$ and $\|G_k^\tau - Y\| \le \|G_k^\tau - G_k\| + \|G_k - Y\|$, it follows from (3.19) and (3.20), by setting $k \leftarrow k+1$, that we prove the assertions in (3.12) and (3.13). ◻

Remark 3.1.

(a) To obtain an approximate solution $H_{\bar{k}}^\tau$ from $\{Q_{1\bar{k}}^\tau, \Sigma_{\bar{k}}^\tau, Q_{2\bar{k}}^\tau\}$ generated by the SDA_ls_ε algorithm will be the most expensive step. Specifically, we need a postprocess for the computation of $H_{\bar{k}}^\tau = Q_{1\bar{k}}^\tau \Sigma_{\bar{k}}^\tau Q_{2\bar{k}}^{\tau^\top}$ which require $O(n^2)$ flops and n^2 memory. Furthermore, the computed $H_{\bar{k}}^\tau$ is no longer nonnegative. If a nonnegative solution is required, it is suggested to set $H_{\bar{k}}^{\tau+} := (H_{\bar{k}}^\tau + |H_{\bar{k}}^\tau|)/2$. Since $X > 0$, it is easily seen that the forward error of $H_{\bar{k}}^{\tau+}$, $\|H_{\bar{k}}^{\tau+} - X\|$, can be estimated by the upper bound of (3.12) in Theorem 3.1.

(b) For $k = \bar{k}$ in (3.12), we see that the coefficients $c_i \equiv 1 + \prod_{j=1}^{\bar{k}-i}(1 + \eta_{\bar{k}-j})$ of ε_i are decreasing, for $i = 1, \ldots, \bar{k}$. It is reasonable to choose the tolerance sequence $\{\varepsilon_i\}_{i=1}^{\bar{k}}$ as an increasing sequence. However, in general, it is hard to estimate those coefficients of ε_i beforehand. Therefore, in practice, we suggest to choose a constant sequence of tolerances $\{\varepsilon_\tau\}_{i=1}^{\bar{k}}$ (e.g., $\varepsilon_\tau = 10^{-3}, 10^{-4}, \ldots, 10^{-15}$). From our numerical experiments, the forward errors of $H_{\bar{k}}^\tau$ and $G_{\bar{k}}^\tau$ almost have the same order of the given truncation tolerance ε_τ.

4. Computational issues.

4.1. Residual and convergence control.

In the SDA_ls_ε, we should compute residuals and differences of iterates carefully in $O(n)$ complexity.

From (3.12), consider the difference of successive iterates

$$\begin{aligned} dH_{k+1}^\tau = H_{k+1}^\tau - H_k^\tau &= Q_{1,k+1}^\tau \Sigma_{k+1}^\tau Q_{2,k+1}^{\tau^\top} - Q_{1k}^\tau \Sigma_k^\tau Q_{2k}^{\tau^\top} \\ &= \left[Q_{1,k+1}^\tau \mid Q_{1k}^\tau \right] (\Sigma_{k+1}^\tau \oplus \Sigma_k^\tau) \begin{bmatrix} Q_{2,k+1}^{\tau^\top} \\ -Q_{2k}^{\tau^\top} \end{bmatrix} \equiv Y_{1k} (\Sigma_{k+1}^\tau \oplus \Sigma_k^\tau) Y_{2k}^\top, \end{aligned}$$

where $Q_{ik},\ Q_{i,k+1}$ $(i = 1, 2)$ are orthogonal. We compute $\|dH_{k+1}^\tau\|$ efficiently as follows:

$$\|dH_{k+1}^\tau\| = \|R_{1k}^h (\Sigma_{k+1}^\tau \oplus \Sigma_k^\tau) R_{2k}^{h\top}\|, \tag{4.1}$$

where $Y_{1k} = W_{1k}^h R_{1k}^h$ and $Y_{2k} = W_{2k}^h R_{2k}^h$ are the QR decompositions of Y_{1k} and Y_{2k}, respectively. Similarly, we have

$$\|dG_{k+1}^\tau\| = \|R_{1k}^g (\Gamma_{k+1}^\tau \oplus \Gamma_k^\tau) R_{2k}^{g\top}\| \tag{4.2}$$

with the QR decompositions $[P_{1,k+1}^\tau, P_{1k}^\tau] = W_{1k}^g R_{1k}^g$ and $[P_{2,k+1}^\tau, -P_{2k}^\tau] = W_{2k}^g R_{2k}^g$.

From the NARE (1.1) we have

$$\begin{aligned} \mathcal{R}(H_k^\tau) &= Q_{1k}^\tau \Sigma_k^\tau \left(Q_{2k}^{\tau^\top} C Q_{1k}^\tau \right) \Sigma_k^\tau Q_{2k}^{\tau^\top} - A Q_{1k}^\tau \Sigma_k^\tau Q_{2k}^{\tau^\top} - Q_{1k}^\tau \Sigma_k^\tau Q_{2k}^{\tau^\top} D + B_1 B_2^\top \\ &= [Q_{1k}^\tau \mid A Q_{1k}^\tau \mid B_1] \begin{bmatrix} \Sigma_k^\tau (Q_{2k}^{\tau^\top} C Q_{1k}^\tau) \Sigma_k^\tau & -\Sigma_k^\tau & 0 \\ -\Sigma_k^\tau & 0 & 0 \\ 0 & 0 & I_m \end{bmatrix} \begin{bmatrix} Q_{2k}^{\tau^\top} \\ Q_{2k}^{\tau^\top} D \\ B_2^\top \end{bmatrix} \\ &\equiv Z_{1k} \Phi_k Z_{2k}^\top. \end{aligned}$$

TABLE 4.1
Operation and memory counts for the kth iteration in Algorithm 1 (SDA_ls_ε).

Computation	Flops	Memory
$\breve{\Sigma}_k^\tau$, $\breve{\Gamma}_k^\tau$	$4l_k m_k n$	–
$F_k^\tau Q_{1k}^\tau$, $F_k^{\tau\top} P_{2k}^\tau$	$\left[2^k(c_\gamma + 4\sum_{j=1}^k 2^{-j} l_j)\right](l_k + m_k)n$	$2n\sum_{j=1}^{k-1} l_j$
$E_k^\tau P_{1k}^\tau$, $E_k^{\tau\top} Q_{2k}^\tau$	$\left[2^k(c_\gamma + 4\sum_{j=1}^k 2^{-j} m_j)\right](l_k + m_k)n$	$2n\sum_{j=1}^{k-1} m_j$
$F_{1,k+1}^\tau$, $E_{1,k+1}^\tau$	$4l_k m_k n$	$2n(m_k + l_k)$
Orthogonalize $F_k^\tau Q_{1k}^\tau$, $F_k^{\tau\top} P_{2k}^\tau$, $E_k^\tau P_{1k}^\tau$ and $E_k^{\tau\top} Q_{2k}^\tau$	$2[6(m_k^2 + l_k^2) + m_k + l_k]n$	–
$\widehat{\Sigma}_{k+1}$, $\widehat{\Gamma}_{k+1}$, Σ_{k+1}^τ, Γ_{k+1}^τ	$O(l_k^3 + m_k^3)$	$l_{k+1} + m_{k+1}$
$Q_{i,k+1}^\tau$, $P_{i,k+1}^\tau$ $(i = 1, 2)$	$8(l_k l_{k+1} + m_k m_{k+1})n$	$2n(l_{k+1} + m_{k+1})$
Total flops & memory	$\Big[2^{k+1}\left(c_\gamma + 2\sum_{j=1}^k 2^{-j}(l_j + m_j)\right)(l_k + m_k)$ $+8(l_k m_k + l_k l_{k+1} + m_k m_{k+1})$ $+12(m_k^2 + l_k^2) + 2(m_k + l_k)\Big]\, n$	$2n\sum_{j=1}^{k+1}(l_j + m_j)$

Then the residual and the relative residual,

$$r_k \equiv \|\mathcal{R}(H_k^\tau)\|, \quad \tilde{r}_k = \frac{r_k}{\|H_k^\tau C H_k^\tau\| + \|H_k^\tau D\| + \|A H_k^\tau\| + \|B\|}, \tag{4.3}$$

can be efficiently calculated by

$$\|\mathcal{R}(H_k^\tau)\| = \|R_{1k}^r \Phi_k R_{2k}^{r\top}\|, \; \|H_k^\tau C H_k^\tau\| = \|\Sigma_k^\tau (Q_{2k}^{\tau\top} C Q_{1k}^\tau)\Sigma_k^\tau\|,$$
$$\|A H_k^\tau\| = \|R_k^a \Sigma_k^\tau\|, \quad \|H_k^\tau D\| = \|\Sigma_k^\tau R_k^{d\top}\|, \quad \|B\| = \|R_{b1} R_{b2}^\top\|,$$

with the QR decompositions $Z_{ik} = W_{ik}^r R_{ik}^r$, $AQ_{1k}^\tau = W_k^a R_k^a$, $D^\top Q_{2k}^\tau = W_k^d R_k^d$, and $B_i = W_i^b R_{bi}$, for $i = 1, 2$.

4.2. Operation and memory counts. We shall assume that $c_\gamma n$ flops ($n = \max\{n_1, n_2\}$) are required in the solution of $\widetilde{M} z = b$ or $\widetilde{M}^\top z = b$ (with $\widetilde{M} = A_\gamma,\ b \in \mathbb{R}^{n_1}$ or $\widetilde{M} = D_\gamma,\ b \in \mathbb{R}^{n_2}$). The operation count for the QR decomposition of an $n \times r$ matrix is $4r^2 n$ flops [11, p. 250]. A start up cost of $(c_1 + c_2 + c_3)n$ flops is made up of the following:

(1) set up $A_\gamma = A + \gamma I_{n_1}$ and $D_\gamma = D + \gamma I_{n_2}$, requiring $n_1 + n_2 \leq 2n$ flops; we shall denote this part of the count by $c_1 n$ flops, with $c_1 = 2$;

(2) set up Q_{i0} and P_{i0} $(i = 1, 2)$ as in (2.2) with the help of (2.3), requiring $c_2 n$ flops with $c_2 = 2\left[(c_\gamma + 1)(m + l) + 2lm + m^2 + l^2\right]$; and

(3) set up Q_{i0}^τ and P_{i0}^τ $(i = 1, 2)$ as in (2.17), requiring $c_3 n$ flops with $c_3 = 12(l^2 + m^2)$.

The operation and memory counts of Algorithm 1 (SDA_ls_ε) for the kth iteration are summarized in Table 4.1. In the third column, the number of variables is recorded. Only $O(n)$ operations or memory requirement are included. Note that most of the work done in the computation of $F_k^\tau Q_{1k}^\tau$, $F_k^{\tau\top} P_{2k}^\tau$, $E_k^\tau P_{1k}^\tau$, and $E_k^{\tau\top} Q_{2k}^\tau$ in (2.23) has to be calculated recursively, as E_k^τ and F_k^τ in (2.21) are not available explicitly.

With l_k and m_k controlled by the compression and truncation in section 2.2, the operation count will be dominated by the calculation of $F_k^\tau Q_{1k}^\tau$, $F_k^{\tau\top} P_{2k}^\tau$, $E_k^\tau P_{1k}^\tau$, and $E_k^{\tau\top} Q_{2k}^\tau$. In our numerical examples in section 5, the flop count near the end of Algorithm 1 dominates, with the work involved in one iteration approximately double that of the previous one. This corresponds to the 2^{k+1} factor in the total flop count.

5. Numerical examples. We constructed the examples as in [23], A and D are rank one updates of nonsingular diagonal matrices and B and C are rank one, generated randomly. Three examples of sizes $n = n_1 = n_2 = 1000, 10000, 100000$ were generated, all satisfying the corresponding solvability conditions. The numerical results in Examples 5.1–5.3 ($n = 1000,\ 10000,\ 100000$) were computed using MATLAB [28] Version R2012b, on an iMac with a 2.97 GHz Intel Core i7 processor and 8 GB RAM, with machine accuracy $eps = 2.22 \times 10^{-16}$.

In Algorithm 1, the stopping criterion is $d_k \equiv \max\{\|dH_k^\tau\|, \|dG_k^\tau\|\} < \varepsilon_c$, where $\|dH_k^\tau\| = \|H_k^\tau - H_{k-1}^\tau\|$ and $\|dG_k^\tau\| = \|G_k^\tau - G_{k-1}^\tau\|$ and convergence tolerance ε_c; please also consult the convergence results in Theorem 3.1. All numerical experiments were considered with a constant truncation tolerance ε_τ in each iteration, i.e., $\varepsilon_i = \varepsilon_\tau$ for $i = 0, 1, \ldots, \bar{k}$.

In Example 5.1 with the smallest $n = 1000$, we apply the SDA (1.5) to compute the near-exact solutions X and Y of NARE (1.1) and its dual equation (1.6). These were then used to illustrate the results for $\text{rank}_\tau(X)$, $\text{rank}_\tau(Y)$ in Table 5.1 and the forward errors in Tables 5.2–5.3. Effects of different tolerances ε_τ (or ε_c) are also presented in Table 5.2 (and in Table 5.3).

In Examples 5.2–5.3, the iterations in the SDA_ls_ε are reported for a corresponding set of tolerances ε_c and ε_τ. In Tables 5.4–5.5 below, $\|H_k^\tau\|$, $\|G_k^\tau\|$, d_k, r_k, $\widetilde{r}_k$, m_k, l_k, δt_k, and t_k are displayed. Note that δt_i is the execution time for the ith iteration and the subtotal execution time $t_k = \sum_{i=1}^k \delta t_i$.

Table 5.1
The numerical ranks of X and Y with respect to various τ.

τ	10^{-3}	10^{-5}	10^{-7}	10^{-9}	10^{-11}	10^{-13}	10^{-15}
$\text{rank}_\tau(X)$	4	7	11	14	17	21	24
$\text{rank}_\tau(Y)$	4	8	11	14	18	21	24

Table 5.2
Numerical results with various truncation tolerances ε_τ.

ε_τ	10^{-3}	10^{-7}	10^{-11}	10^{-15}
$\bar{k}$	12	12	12	12
$m_{\bar{k}}$	3	10	17	24
$l_{\bar{k}}$	3	11	17	24
$\|H_{\bar{k}}^\tau - H_{\bar{k}-1}^\tau\|$	1.208e−13	6.706e−13	6.844e−13	6.844e−13
$\|G_{\bar{k}}^\tau - G_{\bar{k}-1}^\tau\|$	1.767e−12	9.638e−12	9.640e−12	9.640e−12
$r_{\bar{k}}$	1.134e−03	7.820e−08	1.490e−11	4.245e−15
$\|H_{\bar{k}}^\tau - X\|$	1.494e−03	1.473e−07	1.843e−11	7.091e−15
$\|G_{\bar{k}}^\tau - Y\|$	2.174e−03	8.537e−08	1.456e−11	7.691e−15
$\|H_{\bar{k}}^{\tau^-}\|$	1.606e−12	0	0	0
$\|G_{\bar{k}}^{\tau^-}\|$	9.034e−12	0	0	0
$t_{\bar{k}}$	2.104	4.449	6.471	8.377

Table 5.3
Numerical results with various truncation tolerances ε_c.

ε_c	10^{-3}	10^{-7}	10^{-11}	10^{-15}
$\bar{k}$	8	11	12	13
$m_{\bar{k}}$	17	19	19	19
$l_{\bar{k}}$	17	19	19	19
$\Vert H_{\bar{k}}^{\tau} - H_{\bar{k}-1}^{\tau}\Vert$	1.251e−04	7.131e−09	6.844e−13	1.918e−16
$\Vert G_{\bar{k}}^{\tau} - G_{\bar{k}-1}^{\tau}\Vert$	4.982e−04	9.899e−08	9.640e−12	1.311e−16
$r_{\bar{k}}$	1.163e−07	8.310e−13	8.310e−13	8.310e−13
$\Vert H_{\bar{k}}^{\tau} - X\Vert$	1.751e−05	1.094e−12	1.077e−12	1.077e−12
$\Vert G_{\bar{k}}^{\tau} - Y\Vert$	1.417e−04	9.765e−12	1.316e−12	1.316e−12
$\Vert H_{\bar{k}}^{\tau^-}\Vert$	0	0	0	0
$\Vert G_{\bar{k}}^{\tau^-}\Vert$	0	0	0	0
$t_{\bar{k}}$	0.3517	3.227	6.937	12.93

Table 5.4
$n = 10000$, $\varepsilon_\tau = 10^{-12}$, $\varepsilon_c = 10^{-8}$.

k	$\Vert H_k^\tau\Vert$	$\Vert G_k^\tau\Vert$	d_k	r_k	$\widetilde{r}_k$	m_k	l_k	δt_k	t_k
1	0.24203	0.23622	4.245e−2	1.110e−02	2.513e−02	2	2	1.2e−2	1.2e−2
2	0.25704	0.25051	2.643e−2	3.201e−03	7.165e−03	4	4	4.5e−2	5.7e−2
3	0.26182	0.25500	1.388e−2	7.928e−04	1.771e−03	8	8	3.1e−2	8.8e−2
4	0.26294	0.25606	6.634e−3	1.888e−04	4.216e−04	10	10	1.1e−1	1.9e−1
5	0.26316	0.25627	3.166e−3	4.494e−05	1.003e−04	12	12	2.6e−1	4.6e−1
6	0.26320	0.25631	1.473e−3	9.990e−06	2.231e−05	14	14	6.3e−1	1.1e+0
7	0.26321	0.25631	6.167e−4	1.904e−06	4.252e−06	15	15	1.5e+0	2.6e+0
8	0.26321	0.25632	2.365e−4	3.021e−07	6.746e−07	17	17	3.2e+0	5.8e+0
9	0.26321	0.25632	6.564e−5	3.417e−08	7.631e−08	18	18	7.5e+0	1.3e+1
10	0.26321	0.25632	1.076e−5	1.295e−09	2.891e−09	19	19	1.6e+1	3.0e+1
11	0.26321	0.25632	6.023e−7	3.942e−12	8.803e−12	20	20	3.4e+1	6.4e+1
12	0.26321	0.25632	3.142e−9	1.247e−12	2.784e−12	20	20	7.3e+1	1.4e+2

Table 5.5
$n = 100000$, $\varepsilon_\tau = 10^{-12}$, $\varepsilon_c = 10^{-8}$.

k	$\Vert H_k^\tau\Vert$	$\Vert G_k^\tau\Vert$	d_k	r_k	$\widetilde{r}_k$	m_k	l_k	δt_k	t_k
1	0.23862	0.23907	4.180e−02	1.088e−02	2.497e−02	2	2	4.5e−2	4.5e−2
2	0.25331	0.25372	2.587e−02	3.142e−03	7.130e−03	4	4	1.5e−1	1.9e−1
3	0.25797	0.25836	1.378e−02	8.036e−04	1.820e−03	8	8	4.2e−1	6.2e−1
4	0.25910	0.25948	6.832e−03	1.969e−04	4.457e−04	10	10	1.7e+0	2.3e+0
5	0.25933	0.25972	3.309e−03	4.709e−05	1.066e−04	12	12	4.1e+0	6.4e+0
6	0.25938	0.25976	1.567e−03	1.091e−05	2.471e−05	14	14	9.7e+0	1.6e+1
7	0.25939	0.25977	7.140e−04	2.394e−06	5.420e−06	15	15	2.2e+1	3.8e+1
8	0.25939	0.25977	2.961e−04	4.475e−07	1.013e−06	17	17	4.7e+1	8.6e+1
9	0.25939	0.25977	9.725e−05	5.627e−08	1.274e−07	18	18	1.1e+2	1.9e+2
10	0.25939	0.25977	1.896e−05	2.937e−09	6.648e−09	20	20	2.3e+2	4.2e+2
11	0.25939	0.25977	1.276e−06	2.340e−11	5.297e−11	21	21	5.0e+2	9.2e+2
12	0.25939	0.25977	1.146e−08	1.181e−12	2.672e−12	21	21	1.0e+3	2.0e+3
13	0.25939	0.25977	1.662e−12	1.181e−12	2.672e−12	21	21	2.1e+3	4.0e+3

Example 5.1 ($n = 1000$). In this example, we have performed three tests.

Test 1. We first apply the SDA (1.5) with initial (1.4) to compute the near-exact solutions X and Y of NARE (1.1) and its dual equation (1.6). The SDA converges after 12 iterations and the norms ($\Vert X\Vert$, $\Vert Y\Vert$), residuals ($\Vert\mathcal{R}(X)\Vert$, $\Vert\mathcal{D}(Y)\Vert$)

are estimated, respectively, as

$$\|X\| = 2.5748 \times 10^{-1}, \quad \|\mathcal{R}(X)\| = 5.9875 \times 10^{-17},$$
$$\|Y\| = 2.6545 \times 10^{-1}, \quad \|\mathcal{D}(Y)\| = 5.6928 \times 10^{-17}.$$

Table 5.1 shows the $\mathrm{rank}_\tau(X)$ and $\mathrm{rank}_\tau(Y)$ with $\tau = 10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}, 10^{-11}$, 10^{-13}, and 10^{-15}. Note that $\mathrm{rank}_\tau(X)$ and $\mathrm{rank}_\tau(Y)$ are much smaller than the matrix size $n = 1000$.

Test 2. In the test, we set convergence tolerance $\varepsilon_c = 10^{-8}$ and employ Algorithm 1 with various truncation tolerances ε_τ. Suppose that Algorithm 1 converges after $\bar{k}$ iterations, i.e., $d_{\bar{k}} < \varepsilon_c = 10^{-8}$. To determinate whether the computed solutions, $H_{\bar{k}}^\tau$ and $G_{\bar{k}}^\tau$, are nonnegative, we denote

$$H_{\bar{k}}^{\tau^-} = (H_{\bar{k}}^\tau - |H_{\bar{k}}^\tau|)/2, \quad G_{\bar{k}}^{\tau^-} = (G_{\bar{k}}^\tau - |G_{\bar{k}}^\tau|)/2.$$

From the results of Test 1, we have near-exact solutions X and Y. Hence, we can compute the forward errors, $\|H_{\bar{k}}^\tau - X\|$ and $\|G_{\bar{k}}^\tau - Y\|$ in the example. The numerical results are shown in Table 5.2.

Table 5.2 shows that Algorithm 1 converges within 12 iterations (in $2.1 \sim 8.4$ seconds) for various tolerances ε_τ and the residual $r_{\bar{k}}$ and forward errors, $\|H_{\bar{k}}^\tau - X\|$ and $\|G_{\bar{k}}^\tau - Y\|$, are heavily dependent on the chosen truncation tolerances ε_τ. Furthermore, the computed solutions, $H_{\bar{k}}^\tau$ and $G_{\bar{k}}^\tau$, are nonnegative matrices when $\varepsilon_\tau = 10^{-7}, 10^{-11}, 10^{-15}$.

Test 3. In the test, we set truncation tolerance $\varepsilon_\tau = 10^{-12}$ and employ Algorithm 1 with various convergence tolerances ε_c. The numerical results are shown in Table 5.3.

Table 5.3 shows that Algorithm 1 converges within $8 \sim 13$ iterations for various tolerances ε_c and the residual and forward errors achieve the accuracy of $O(\varepsilon_\tau)$ for $\varepsilon_c = 10^{-7}, 10^{-11}, 10^{-15}$.

Example 5.2 ($n = 10000$). In this example, we set the truncation tolerance $\varepsilon_\tau = 10^{-12}$ and convergence tolerance $\varepsilon_c = 10^{-8}$. In Table 5.4, the residual (or relative residual) achieves the accuracy of $O(\varepsilon_\tau)$ within 12 iterations, in 140 seconds (execution time).

Example 5.3 ($n = 100000$). In this example, we set the truncation tolerance $\varepsilon_\tau = 10^{-12}$ and convergence tolerance $\varepsilon_c = 10^{-8}$. In Table 5.5, the residual (or relative residual) achieves the accuracy of $O(\varepsilon_\tau)$ within 13 iterations, in 4000 seconds (execution time).

6. Conclusions.

We have proposed a structure-preserving doubling algorithm for the large-scale nonsymmetric algebraic Riccati equation (1.1), the SDA_ls_ε, with A and D being large and sparse(like), and B and C being low ranked. We apply the SMWF when appropriate and do not form E_k and F_k (the iterates for E and F) explicitly. For well-behaved NAREs, low-ranked approximations to the solutions X and Y can be obtained efficiently. The convergence of the SDA_ls_ε is quadratic, ignoring the compression and truncation of $Q_{i,k}$ and $P_{i,k}$, as shown in [17, 25]. The computational complexity and memory requirement are both $O(n)$, provided that the growth in the dimensions of $Q_{i,k}$ and $P_{i,k}$ is controlled. In the error analysis part, we gave a first order forward error bound for the computed approximate solution in Theorem 3.1. Notice that large-scale NAREs, arisen naturally from transport theory [22, 23], have not been investigated before. Our technique can be applied when A

and D are large and sparse(like), or are products (inverses) of such matrices. The feasibility of the SDA_ls_ε depends on whether $A^{-1}u$, $A^{-\top}u$, $D^{-1}v$, and $D^{-\top}v$ can be formed efficiently, for arbitrary vectors u and v.

Acknowledgments. Parts of this project were completed while the first author visited Monash University and when the second author visited the CMMSC and the NCTS at the National Chiao Tung University, and we would like to acknowledge the support from these universities.

We also would like to thank Mr. Chang-Yi Weng for his valuable suggestions on the beginning version. Last but not least, we would like to thank the referees for their valuable comments.

REFERENCES

[1] Z.-Z. Bai, Y.-H. Gao, and L.-Z. Lu, *Fast iterative schemes for nonsymmetric algebraic Riccati equations arising from transport theory*, SIAM J. Sci. Comput., 30 (2008), pp. 804–818.

[2] D.S. Bernstein and W.M. Haddad, *LQG control with an H_∞ performance bound: A Riccati equation approach*, IEEE Trans. Automat. Control, 34 (1989), pp. 293–305.

[3] D.A. Bini, B. Iannazzo, and F. Poloni, *A fast Newtons 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, *Transforming algebraic Riccati equations into unilateral quadratic matrix equations*, Numer. Math., 116 (2010), pp. 553–578.

[5] C.-Y. Chiang, E.K.-W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, and S.-F. Xu, *Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case*, SIAM J. Matrx Anal. Appl., 31 (2009), pp. 227–247.

[6] E.K.-W. Chu, H.-Y. Fan, and W.-W. Lin, *A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations*, Linear Algebra Appl., 396 (2005), pp. 55–80.

[7] E.K.-W. Chu, H.-Y. Fan, W.-W. Lin, and C.-S. Wang, *A structure-preserving doubling algorithm for periodic discrete-time algebraic Riccati equations*, Internat. J. Control, 77 (2004), pp. 767–788.

[8] E.K.-W. Chu, T.M. Huang, W.-W. Lin, and C.-T. Wu, *Palindromic eigenvalue problems: A brief survey*, Taiwanese J. Math., 14 (2010), pp. 743–779.

[9] B. de Moor and J. David, *Total linear least squares and the algebraic Riccati equations*, Systems Control Lett., 18 (1992), pp. 329–337.

[10] I. Gohberg and M.A. Kaashoek, *An inverse spectral problem for rational matrix functions and minimal divisibility*, Integral Equations Operator Theory, 10 (1987), pp. 437–465.

[11] G.H. Golub and C.F. van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

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

[13] C.-H. Guo, *A new class of nonsymmetric algebraic Riccati equations*, Linear Algebra Appl., 426 (2007), pp. 636–649.

[14] C.-H. Guo and N.J. Higham, *Iterative solution of a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 396–412.

[15] C.-H. Guo and P. Lancaster, *Analysis and modification of Newton's method for algebraic Riccati equations*, Math. Comp., 67 (1998), pp. 1089–1105.

[16] 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.

[17] X.-X. Guo, W.-W. Lin, and S.-F. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equations*, Numer. Math., 103 (2006), pp. 393–412.

[18] D. Hinrichsen, B. Kelb, and A. Linnemann, *An algorithm for the computation of the structured complex stability radius*, Automatica J. IFAC, 25 (1989), pp. 771–775.

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

[20] P. Lancaster and L. Rodman, *Solutions of the continuous and discrete-time algebraic Riccati equations: A review*, in The Riccati Equations, S. Bittanti, A.J. Lamb, and J.C. Willems, eds., Springer-Verlag, Berlin, 1991.

[21] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon Press, Oxford, 1995.

[22] T. Li, E.K.-W. Chu, J. Juang, and W.-W. Lin, *Solution of a nonsymmetric algebraic Riccati equation from a two-dimensional transport model*, Linear Algebra Appl., 434 (2011), pp. 201–214.

[23] T. Li, E.K.-W. Chu, J. Juang, and W.-W. Lin, *Solution of a nonsymmetric algebraic Riccati equation from a one-dimensional multi-state transport model*, IMA J. Numer. Anal., 31 (2011), pp. 1453–1467.

[24] T. Li, E.K.-W. Chu, W.-W. Lin, and C.-Y. Weng, *Solving large-scale continuous-time algebraic Riccati equations by doubling*, J. Comput. Appl. Math., 237 (2013), pp. 373–383.

[25] W.-W. Lin and S.-F. Xu, *Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations*, SIAM J. Matrix Anal. Appl., 28 (2008), pp. 26–39.

[26] Y. Lin, L. Bao, and Y. Wei, *A modified Newton method for solving non-symmetric algebraic Riccati equations arising in transport theory*, IMA J. Numer. Anal., 29 (2008), pp. 215–224.

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

[28] Mathworks, *MATLAB User's Guide*, The MathWorks, Natwick, MA, 1992.

[29] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem*, Lecture Notes in Control and Information Sciences 163, Springer-Verlag, Berlin, 1991.

[30] V. Mehrmann and H. Xu, *Explicit solutions for a Riccati equation from transport theory*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1339–1357.

[31] W.-G. Wang, W.-C. Wang, and R.-C. Li, *Alternating-directional doubling algorithm for M-matrix algebraic Riccati equations*, SIAM. J. Matrix Anal. Appl., 33 (2012), pp. 170–194.

[32] D. Williams, *A potential-theoretical note on the quadratic Wiener-Hopf equation for Q-matrices*, in Seminar on Probability XVI, Lecture Notes in Mathematics 920, Springer-Verlag, Berlin, 1982, pp. 91–94.