



Article

Factorized Doubling Algorithm for Large-Scale High-Ranked Riccati Equations in Fractional System

Bo Yu and Ning Dong *

School of Science, Hunan University of Technology, Zhuzhou 412007, China; yubo@hut.edu.cn

* Correspondence: dongning@hut.edu.cn

Abstract: In real-life control problems, such as power systems, there are large-scale high-ranked discrete-time algebraic Riccati equations (DAREs) from fractional systems that require stabilizing solutions. However, these solutions are no longer numerically low-rank, which creates difficulties in computation and storage. Fortunately, the potential structures of the state matrix in these systems (e.g., being banded-plus-low-rank) could be beneficial for large-scale computation. In this paper, a factorized structure-preserving doubling algorithm (FSDA) is developed under the assumptions that the non-linear and constant terms are positive semidefinite and banded-plus-low-rank. The detailed iteration scheme and a deflation process for FSDA are analyzed. Additionally, a technique of partial truncation and compression is introduced to reduce the dimensions of the low-rank factors. The computation of residual and the termination condition of the structured version are also redesigned. Illustrative numerical examples show that the proposed FSDA outperforms SDA with hierarchical matrices toolbox (SDA_HODLR) on CPU time for large-scale problems.

Keywords: large-scale Riccati equations; high-ranked terms; deflation; partial truncation and compression; doubling algorithm



Citation: Yu, B.; Dong, N. Factorized Doubling Algorithm for Large-Scale High-Ranked Riccati Equations in Fractional System. *Fractal Fract.* **2023**, *7*, 468. <https://doi.org/10.3390/fractalfract7060468>

Academic Editor: António Lopes

Received: 6 May 2023

Revised: 7 June 2023

Accepted: 8 June 2023

Published: 10 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Consider the fractional system [1,2]

$$\Delta^{(\alpha)}x(t+1) = \mathbb{A}x(t) + \mathbb{B}u(t), \quad y(t) = Cx(t), \quad (1)$$

where $\alpha \in (0, 1)$ and (α) represents the order of the fractional derivative, $\mathbb{A} \in \mathbb{R}^{N \times N}$, $\mathbb{B} \in \mathbb{R}^{N \times m}$ and $C \in \mathbb{R}^{l \times N}$ with $m, l \leq N$. If $\Delta^{(\alpha)}x(t+1)$ is approximated by the Grünwald–Letnikov rule [3] at $k = 1$, the system (1) is equivalent to the discrete-time linear system

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \quad (2)$$

where $A = h^\alpha \mathbb{A} + \alpha I$ and $B = h^\alpha \mathbb{B}$. The corresponding optimal control and the feedback gain can be expressed in terms of the unique positive semidefinite stabilizing solution of the discrete-time algebraic Riccati Equation (DARE)

$$\mathcal{D}(X) \equiv -X + A^\top X(I + GX)^{-1}A + H = 0, \quad A, G, H \in \mathbb{R}^{N \times N}. \quad (3)$$

There have been numerous methods, including classical and state-of-the-art techniques, developed over the past few decades to solve this equation in a numerically stable manner. See [4–15] and the references therein for more details.

In many large-scale control problems, the matrix $G = BR^{-1}B^\top$ in the non-linear term and $H = C^\top T^{-1}C$ in the constant term are of low-rank with $B \in \mathbb{R}^{N \times m^g}$, $R \in \mathbb{R}^{m^g \times m^g}$, $C \in \mathbb{R}^{m^h \times N}$, $T \in \mathbb{R}^{m^h \times m^h}$, and $m^g, m^h \ll N$. Then the unique positive definite stabilizing solution in the DARE (3) or its dual equation can be approximated numerically by a low-rank matrix [16,17]. However, when the constant term H in the DARE equation has a

high-rank structure, the stabilizing solution is no longer numerically low-ranked, making its storage and outputting difficult. To solve this issue, an adapted version of the doubling algorithm, named SDA_h, was proposed in [18]. The main idea behind SDA_h is to take advantage of the numerical low-rank of the stabilizing solution in the dual equation to estimate the residual of the original DARE. In this way, SDA_h can efficiently evaluate the residual and output the feedback gain. An interesting question up to now is:

Can SDA solve the large-scale DAREs efficiently when both G and H are of high-rank?

The main difficulty, in this case, lies in that the stabilizing solutions both in DARE (3) and its dual equation are not of low-rank, making the direct application of SDA difficult for large-scale problems, especially the estimation of residuals and the realization of algorithmic termination. This paper attempts to overcome this obstacle. Rather than answering the above question completely, DARE (3) with the banded-plus-low-rank structure

$$A = D^A + L_{10}^A K^A (L_{20}^A)^\top \quad (4)$$

is considered, where $D^A \in \mathbb{R}^{N \times N}$ is a banded matrix, $L_{10}^A, L_{20}^A \in \mathbb{R}^{N \times m^a}$ are low-rank matrices and $K^A \in \mathbb{R}^{m^a \times m^a}$ is the kernel matrix with $m^a \ll N$. The assumption of (4) is not necessary when G and H are of low rank, i.e., in that case A is allowed to be any (sparse) matrix. We also assume that the high-rank non-linear item and the constant item are of the form

$$G = D^G + L^G K^G (L^G)^\top, \quad H = D^H + L^H K^H (L^H)^\top, \quad (5)$$

where $D^G, D^H \in \mathbb{R}^{N \times N}$ are positive semidefinite banded matrices, $L^G \in \mathbb{R}^{N \times m^g}$, $L^H \in \mathbb{R}^{N \times m^h}$, $K^G \in \mathbb{R}^{m^g \times m^g}$ and $K^H \in \mathbb{R}^{m^h \times m^h}$ are symmetric and $m^g, m^h \ll N$ (here m^g and m^h might be zero). In addition, we assume that D^A, D^G , and D^H are all banded matrices with banded inverse (BMBI), which has some applications in the power system [19–21]. See also [22–29], as well as their references for other applications.

The main contributions in this paper are:

- Although the hierarchical (e.g., HODLR) structure [30,31] can be employed to run the SDA to cope with large-scale DAREs with both high-rank H and G , it is the first to develop SDA to the factorized form—FSDA—to deal with such DAREs.
- The structure of the FSDA iterative sequence is explicitly revealed to consist of two parts—the banded part and the low-rank part. The banded part can iterate independently while the low-rank part relies heavily on the product of the banded part and the low-rank part.
- A deflation process of the low-rank factors is proposed to reduce the column number of the low-rank part. The conventional truncation and compression in [17,18] for the whole low-rank factor does not work as it destroys the implicit structure and makes the subsequent deflation infeasible. Instead, a partial truncation and compression (PTC) technique is then devised to impose merely on the exponentially increasing part (after deflation), effectively slimming the dimensions of the low-rank factors.
- The termination criterion of FSDA consists of two parts. The residual of the banded part is considered in the pre-termination, and only if it is small enough, the actual termination criterion involving the low-rank factors is computed. This way, the time-consuming detection of the terminating condition is reduced in complexity.

The research in this field is also motivated by other applications, such as the finite element methods (FEM). In FEM, the matrices resulting from discretizing the matrix equations exhibit a sparse and structured pattern [32,33]. By capitalizing on these advantages, iterative methods designed for such matrices can significantly enhance computational efficiency, minimize memory usage, and lead to quicker solutions for large-scale problems.

The whole paper is organized as follows. Section 2 describes the FSDA for DAREs (3) with high-rank non-linear and constant terms. The deflation process for the low-rank factors and kernels is given in Section 3. Section 4 dwells on the technique of PTC to slim the dimensions of low-rank factors and kernels. The way to compute the residual, as well as

the concrete implementation of the FSDA, is described in Section 5. Numerical experiments are listed in Section 6 to show the effectiveness of the FSDA.

Notation 1. I_N (or simply I) is the $N \times N$ identity matrix. For a matrix $A \in \mathbb{R}^{N \times N}$, $\rho(A)$ denotes the spectral radius of A . For symmetric matrices A and $B \in \mathbb{R}^{N \times N}$, we say $A > B$ ($A \geq B$) if $A - B$ is a positive definite (semi-definite) matrix. Unless stated otherwise, the norm $\|\cdot\|$ is the F-norm of a matrix. For a sequence of matrices $\{A_i\}_{i=1}^k$, $\prod_{i=k}^0 A_i = A_k A_{k-1} \dots A_1 A_0$. For a banded matrix B , $bw(B)$ represents the bandwidth. Additionally, the Sherman–Morrison–Woodbury (SMW) formula (see [34] for example), $(M + UDV^\top)^{-1} = M^{-1} - M^{-1}U(D^{-1} + V^\top M^{-1}U)^{-1}V^\top M^{-1}$ is required in the analysis of iterative scheme.

2. SDA and the Structured Iteration for DARE

For DARE

$$\mathcal{D}(X) = -X + A^\top X(I + GX)^{-1}A + H = 0$$

and its dual equation

$$\mathcal{D}_a(Y) = -Y + AY(I + HY)^{-1}A^\top + G = 0, \tag{6}$$

SDA [7] generates a sequence of matrices, for $k \geq 1$

$$\begin{cases} G_k &= G_{k-1} + A_{k-1}(I + G_{k-1}H_{k-1})^{-1}G_{k-1}A_{k-1}^\top, \\ H_k &= H_{k-1} + A_{k-1}^\top H_{k-1}(I + G_{k-1}H_{k-1})^{-1}A_{k-1}, \\ A_k &= A_{k-1}(I + G_{k-1}H_{k-1})^{-1}A_{k-1}, \end{cases} \tag{7}$$

with $A_0 = A$, $G_0 = G$, $H_0 = H$. Under some conditions (see also Theorem 1), $\{A_k\}$ converges to the zero matrix and $\{H_k\}$ and $\{G_k\}$ converge to the stabilizing solutions of $\mathcal{D}(X) = 0$ and $\mathcal{D}_a(Y) = 0$, respectively.

2.1. FSDA for High-Rank Terms

Given banded matrices $D_0^A = D^A$, $D_0^G = D^G$ and $D_0^H = D^H$, low-rank matrices L_{10}^G , L_{10}^A , L_0^H , and L_{20}^A , and kernels $K_0^A = K^A$, $K_0^G = K^G$, and $K_0^H = K^H$ in the structured initial matrices (4) and (5), the FSDA is described inductively as follows, where

$$A_k = D_k^A + L_{1,k}^A K_k^A (L_{2,k}^A)^\top, \quad G_k = D_k^G + L_k^G K_k^G (L_k^G)^\top, \quad H_k = D_k^H + L_k^H K_k^H (L_k^H)^\top \tag{8}$$

with sparse banded matrices $D_k^A, D_k^G, D_k^H \in \mathbb{R}^{N \times N}$, low-rank factors $L_{1,k}^A \in \mathbb{R}^{N \times m_k^{a1}}$, $L_{2,k}^A \in \mathbb{R}^{N \times m_k^{a2}}$, $L_k^G \in \mathbb{R}^{N \times m_k^g}$, $L_k^H \in \mathbb{R}^{N \times m_k^h}$, kernel matrices $K_k^A \in \mathbb{R}^{m_k^{a1} \times m_k^{a2}}$, $K_k^G \in \mathbb{R}^{m_k^g \times m_k^g}$, $K_k^H \in \mathbb{R}^{m_k^h \times m_k^h}$ and $m_k^{a1}, m_k^{a2}, m_k^g, m_k^h \ll N$. Without loss of generality, we assume that $m_0^{a1} = m_0^{a2} \equiv m^a$ and $K_0^A = I_{m^a}$. Otherwise, $L_{20}^A := L_{20}^A (K_0^A)^\top$ and $K_0^A := I_{m^a}$ fulfill the assumption.

We first elaborate the concrete format of banded parts and low-rank factors for $k = 1$ and $k \geq 2$. Note that banded parts are capable of iterating independently, regardless of the low-rank parts and kernels.

Case for $k = 1$.

In the first step, we will assume that $G_0 = D_0^G$ and $H_0 = D_0^H$, i.e., these matrices have no low-rank part. Note that this is only performed in order to simplify exposition. The fully general case with non-trivial low-rank parts will be shown in the case $k \geq 2^i$.

Insert the initial matrices D_0^A, D_0^G , and D_0^H and low-rank matrices L_{10}^A and L_{20}^A into SDA (7). It follows from the SMW formula that

$$\begin{aligned} D_1^G &= D_0^G + D_0^{AGHG} (D_0^A)^\top, \\ D_1^H &= D_0^H + D_0^{A^\top HGH} D_0^A, \\ D_1^A &= D_0^{AGH} D_0^A = D_0^A (D_0^{A^\top HG})^\top \end{aligned} \tag{9}$$

with

$$D_0^{AGHG} = D_0^A(I_N + D_0^G D_0^H)^{-1} D_0^G, \quad D_0^{A^\top HGH} = (D_0^A)^\top (I_N + D_0^H D_0^G)^{-1} D_0^H, \\ D_0^{AGH} = D_0^A(I_N + D_0^G D_0^H)^{-1}, \quad D_0^{A^\top HG} = (D_0^A)^\top (I_N + D_0^H D_0^G)^{-1}.$$

It follows from [35] (Lem 4.5) that the iteration (9) is well defined if D_0^G and D_0^H are both positive semidefinite.

The low-rank factors in (8) are

$$L_1^G = [L_{10}^A, D_0^{AGHG} L_{20}^A], \quad L_1^H = [L_{20}^A, D_0^{A^\top HGH} L_{10}^A], \\ L_{11}^A = [L_{10}^A, D_0^{AGH} L_{10}^A], \quad L_{21}^A = [L_{20}^A, D_0^{A^\top HG} L_{20}^A] \tag{10}$$

and the kernels in the low-rank parts are

$$K_1^G = \begin{bmatrix} (L_{20}^A)^\top D_0^{GHG} L_{20}^A & I_{m_0^g} \\ I_{m_0^g} & 0 \end{bmatrix}, \quad K_1^H = \begin{bmatrix} (L_{10}^A)^\top D_0^{HGH} L_{10}^A & I_{m_0^h} \\ I_{m_0^h} & 0 \end{bmatrix}, \tag{11}$$

$$K_1^A = \begin{bmatrix} (L_{20}^A)^\top D_0^{GH} L_{10}^A & I_{m_0^g} \\ I_{m_0^h} & 0 \end{bmatrix} \tag{12}$$

with

$$D_0^{GHG} = (I_N + D_0^G D_0^H)^{-1} D_0^G, \quad D_0^{HGH} = (I_N + D_0^H D_0^G)^{-1} D_0^H, \quad D_0^{GH} = (I_N + D_0^G D_0^H)^{-1}$$

and $m_0^g = m^a, m_0^h = m^a$.

Case for general $k \geq 2$.

By inserting the banded matrices D_{k-1}^G, D_{k-1}^H and D_{k-1}^A and the low-rank factors $L_{k-1}^G, L_{k-1}^H, L_{1,k-1}^A$ and $L_{2,k-1}^A$ and the kernels K_{k-1}^G, K_{k-1}^H and K_{k-1}^A into SDA (7), banded matrices at the k -th iteration are

$$D_k^G = D_{k-1}^G + D_{k-1}^{AGHG} (D_{k-1}^A)^\top, \\ D_k^H = D_{k-1}^H + D_{k-1}^{A^\top HGH} D_{k-1}^A, \\ D_k^A = D_{k-1}^{AGH} D_{k-1}^A = D_{k-1}^A (D_{k-1}^{A^\top HG})^\top \tag{13}$$

with

$$D_{k-1}^{AGHG} = D_{k-1}^A(I_N + D_{k-1}^G D_{k-1}^H)^{-1} D_{k-1}^G, \quad D_{k-1}^{A^\top HGH} = (D_{k-1}^A)^\top (I_N + D_{k-1}^H D_{k-1}^G)^{-1} D_{k-1}^H, \\ D_{k-1}^{AGH} = D_{k-1}^A(I_N + D_{k-1}^G D_{k-1}^H)^{-1}, \quad D_{k-1}^{A^\top HG} = (D_{k-1}^A)^\top (I_N + D_{k-1}^H D_{k-1}^G)^{-1}.$$

The corresponding low-rank factors are

$$L_k^G = \begin{bmatrix} m_{k-1}^g & m_{k-1}^{a1} & m_{k-1}^g & m_{k-1}^h & m_{k-1}^{a2} \\ L_{k-1}^G & L_{1,k-1}^A & D_{k-1}^{AGH} L_{k-1}^G & D_{k-1}^{AGHG} L_{k-1}^H & D_{k-1}^{AGHG} L_{2,k-1}^A \end{bmatrix} N, \tag{14}$$

$$L_{1,k}^A = \begin{bmatrix} m_{k-1}^{a1} & m_{k-1}^g & m_{k-1}^h & m_{k-1}^{a1} \\ L_{1,k-1}^A & D_{k-1}^{AGH} L_{k-1}^G & D_{k-1}^{AGHG} L_{k-1}^H & D_{k-1}^{AGH} L_{1,k-1}^A \end{bmatrix} N, \tag{15}$$

$$L_k^H = \begin{bmatrix} m_{k-1}^h & m_{k-1}^{a2} & m_{k-1}^g & m_{k-1}^{a1} \\ L_{k-1}^H & L_{2,k-1}^A & D_{k-1}^{A^\top HG} L_{k-1}^H & D_{k-1}^{A^\top HGH} L_{k-1}^G & D_{k-1}^{A^\top HGH} L_{1,k-1}^A \end{bmatrix} N, \tag{16}$$

$$L_{2,k}^A = \begin{bmatrix} m_{k-1}^{a2} & m_{k-1}^g & m_{k-1}^h & m_{k-1}^{a2} \\ L_{2,k-1}^A & D_{k-1}^{A^\top HG} L_{k-1}^H & D_{k-1}^{A^\top HGH} L_{k-1}^G & D_{k-1}^{A^\top HG} L_{2,k-1}^A \end{bmatrix} N. \tag{17}$$

To express the kernels explicitly, let

$$\begin{aligned} \Theta_{k-1}^H &= (L_{k-1}^H)^\top D_{k-1}^{GHG} L_{k-1}^H, & \Theta_{k-1}^G &= (L_{k-1}^G)^\top D_{k-1}^{HGH} L_{k-1}^G, \\ \Theta_{k-1}^{HG} &= (L_{k-1}^H)^\top D_{k-1}^{GH} L_{k-1}^G, & \Theta_{k-1}^A &= (L_{2,k-1}^A)^\top D_{k-1}^{GH} L_{1,k-1}^A, \\ \Theta_{1,k-1}^A &= (L_{1,k-1}^A)^\top D_{k-1}^{HGH} L_{1,k-1}^A, & \Theta_{2,k-1}^A &= (L_{2,k-1}^A)^\top D_{k-1}^{GHG} L_{2,k-1}^A \end{aligned}$$

and

$$\begin{aligned} \Theta_{1,k-1}^{AH} &= (L_{1,k-1}^A)^\top D_{k-1}^{HG} L_{k-1}^H, & \Theta_{1,k-1}^{AG} &= (L_{1,k-1}^A)^\top D_{k-1}^{HGH} L_{k-1}^G, \\ \Theta_{2,k-1}^{AH} &= (L_{2,k-1}^A)^\top D_{k-1}^{GHG} L_{k-1}^H, & \Theta_{2,k-1}^{AG} &= (L_{2,k-1}^A)^\top D_{k-1}^{GH} L_{k-1}^G \end{aligned}$$

with

$$\begin{aligned} D_{k-1}^{GHG} &= (I_N + D_{k-1}^G D_{k-1}^H)^{-1} D_{k-1}^G, & D_{k-1}^{HGH} &= (I_N + D_{k-1}^H D_{k-1}^G)^{-1} D_{k-1}^H, \\ D_{k-1}^{GH} &= (I_N + D_{k-1}^G D_{k-1}^H)^{-1}, & D_{k-1}^{HG} &= (I_N + D_{k-1}^H D_{k-1}^G)^{-1}. \end{aligned}$$

Define the kernel components

$$K_{k-1}^{GH} = \begin{bmatrix} 0 & K_{k-1}^G \\ K_{k-1}^H & 0 \end{bmatrix} \left(I_{m_{k-1}^h + m_{k-1}^s} + \begin{bmatrix} -\Theta_{k-1}^H & \Theta_{k-1}^{HG} \\ (\Theta_{k-1}^{HG})^\top & \Theta_{k-1}^G \end{bmatrix} \begin{bmatrix} -K_{k-1}^H & 0 \\ 0 & K_{k-1}^G \end{bmatrix} \right)^{-1}, \tag{18}$$

$$K_{k-1}^{GHG} = K_{k-1}^{GH} \begin{bmatrix} 0 & I_{m_{k-1}^h} \\ -I_{m_{k-1}^s} & 0 \end{bmatrix}, \quad K_{k-1}^{HGH} = \begin{bmatrix} 0 & -I_{m_{k-1}^h} \\ I_{m_{k-1}^s} & 0 \end{bmatrix} K_{k-1}^{GH} \tag{19}$$

and

$$\begin{aligned} K_{k-1}^{AGHG} &= -K_{k-1}^A [\Theta_{2,k-1}^{AG}, \Theta_{2,k-1}^{AH}] K_{k-1}^{GHG}, \\ K_{k-1}^{A^\top HGH} &= -(K_{k-1}^A)^\top [\Theta_{1,k-1}^{AH}, \Theta_{1,k-1}^{AG}] K_{k-1}^{HGH}, \\ K_{k-1}^{AGHGA^\top} &= K_{k-1}^A \Theta_{2,k-1}^A (K_{k-1}^A)^\top + K_{k-1}^{AGHG} [\Theta_{2,k-1}^{AG}, \Theta_{2,k-1}^{AH}]^\top (K_{k-1}^A)^\top, \\ K_{k-1}^{A^\top HGHA} &= (K_{k-1}^A)^\top \Theta_{1,k-1}^A K_{k-1}^A + K_{k-1}^{A^\top HGH} [\Theta_{1,k-1}^{AH}, \Theta_{1,k-1}^{AG}]^\top K_{k-1}^A, \\ K_{k-1}^{AGH} &= -K_{k-1}^A [\Theta_{2,k-1}^{AG}, \Theta_{2,k-1}^{AH}] K_{k-1}^{GH}, \\ K_{k-1}^{A^\top GH} &= -(K_{k-1}^A)^\top [\Theta_{1,k-1}^{AH}, \Theta_{1,k-1}^{AG}] (K_{k-1}^{GH})^\top, \\ K_{k-1}^{AGHA} &= K_{k-1}^A \Theta_{k-1}^A K_{k-1}^A + K_{k-1}^{AGH} [\Theta_{1,k-1}^{AH}, \Theta_{1,k-1}^{AG}]^\top K_{k-1}^A. \end{aligned} \tag{20}$$

Then the kernel matrices corresponding to $L_k^G, L_k^H,$ and $L_{1,k}^A (L_{2,k}^A)$ at the k -th step are

$$K_k^G = \begin{bmatrix} m_{k-1}^s & m_{k-1}^{a_1} & m_{k-1}^s + m_{k-1}^h & m_{k-1}^{a_2} \\ K_{k-1}^G & 0 & 0 & 0 \\ 0 & K_{k-1}^{AGHGA^\top} & K_{k-1}^{AGHG} & K_{k-1}^A \\ 0 & (K_{k-1}^{AGHG})^\top & -K_{k-1}^{GHG} & 0 \\ 0 & (K_{k-1}^A)^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} m_{k-1}^s \\ m_{k-1}^{a_1} \\ m_{k-1}^s + m_{k-1}^h \\ m_{k-1}^{a_2} \end{bmatrix}, \tag{21}$$

$$K_k^H = \begin{bmatrix} m_{k-1}^h & m_{k-1}^{a_2} & m_{k-1}^h + m_{k-1}^s & m_{k-1}^{a_1} \\ K_{k-1}^H & 0 & 0 & 0 \\ 0 & K_{k-1}^{A^\top HGHA} & K_{k-1}^{A^\top HGH} & (K_{k-1}^A)^\top \\ 0 & (K_{k-1}^{A^\top HGH})^\top & -K_{k-1}^{HGH} & 0 \\ 0 & K_{k-1}^A & 0 & 0 \end{bmatrix} \begin{bmatrix} m_{k-1}^h \\ m_{k-1}^{a_2} \\ m_{k-1}^h + m_{k-1}^s \\ m_{k-1}^{a_1} \end{bmatrix} \tag{22}$$

and

$$K_k^A = \begin{bmatrix} m_{k-1}^{a_2} & m_{k-1}^s + m_{k-1}^h & m_{k-1}^{a_2} \\ K_{k-1}^{AGHA} & K_{k-1}^{AGH} & K_{k-1}^A \\ (K_{k-1}^{A^1GH})^\top & -K_{k-1}^{GH} & 0 \\ K_{k-1}^A & 0 & 0 \end{bmatrix} \begin{matrix} m_{k-1}^{a_1} \\ m_{k-1}^s + m_{k-1}^h \\ m_{k-1}^{a_1} \end{matrix}. \tag{23}$$

Remark 1. 1. The banded parts in (13) in the FSDA can iterate independently of the low-rank parts, motivating to the pre-termination criterion in Section 5.

2. Low-rank factors in (14)–(17) are seen growing in dimension on a scale of $O(4^k)$, obviously intolerable for large-scale problems. So a deflation process and a truncation and compression technique are required to reduce the dimensions of the low-rank factors.

3. In real implementations, low-rank factors and kernels for $k \geq 2$ are actually deflated, truncated, and compressed, as described in the next two sections, where a superscript “dt” is added to the upper right corner of each low-rank factor. Correspondingly, column numbers m_{k-1}^s , m_{k-1}^h , $m_{k-1}^{a_1}$, and $m_{k-1}^{a_2}$ are the ones after deflation, truncation, and compression. Here, we temporarily omit this superscript “dt” just for the convenience when describing the successive iteration process.

2.2. Convergence and the Evolution of the Bandwidth

To obtain the convergence, we further assume that

$$[A, G] \text{ is d-stabilizable and } [H, A] \text{ is d-detectable} \tag{24}$$

and

$$[D^A, D^G] \text{ is d-stabilizable and } [D^H, D^A] \text{ is d-detectable.} \tag{25}$$

The following theorem concludes the convergence of SDA (7), see [35] (Thm 4.3, Thm 4.6) or [36] (Thm 3.1).

Theorem 1. Under the assumption (24), there are unique symmetric positive semi-definite and stabilizing solutions X_s and Y_s to DARE (3) and its dual Equation (6), respectively. Moreover, the sequences $\{G_k\}$, $\{H_k\}$ and $\{A_k\}$ generated by SDA (7) satisfy $0 \leq H \leq H_k \leq H_{k+1} \leq X_s$, $0 \leq G \leq G_k \leq G_{k+1} \leq Y_s$ for all k and

$$\lim_{k \rightarrow \infty} H_k = X_s, \quad \lim_{k \rightarrow \infty} G_k = Y_s, \quad \lim_{k \rightarrow \infty} A_k = 0, \tag{26}$$

all quadratically.

For the banded iterations (9) and (13), we have the following corollary.

Corollary 1. Under the assumption (25), there are unique symmetric positive semi-definite and stabilizing solutions D^X and D^Y to the equation

$$-X + (D^A)^\top X(I + D^G X)^{-1} D^A + D^H = 0 \tag{27}$$

and its dual equation

$$-Y + D^A Y(I + D^H Y)^{-1} (D^A)^\top + D^G = 0 \tag{28}$$

respectively. Moreover, the sequences $\{D_k^G\}$, $\{D_k^H\}$ and $\{D_k^A\}$ generated by iterations (9) and (13) satisfy $0 \leq D^H \leq D_k^H \leq D_{k+1}^H \leq D^X$, $0 \leq D^G \leq D_k^G \leq D_{k+1}^G \leq D^Y$ for all k and

$$\lim_{k \rightarrow \infty} D_k^H = D^X, \quad \lim_{k \rightarrow \infty} D_k^G = D^Y, \quad \lim_{k \rightarrow \infty} D_k^A = 0, \tag{29}$$

all quadratically.

Proof. This is a direct application of Theorem 1 to Equation (27) and its dual Equation (28) under the assumption (25). \square

Corollary 2. Under the conditions of Theorem 1 and Corollary 1, the symmetric positive semidefinite solutions X_s and Y_s to DARE (3) and its dual equation have the decompositions

$$X_s = D^X + L_{lr}^X \text{ and } Y_s = D^Y + L_{lr}^Y.$$

Moreover, for the sequences generated by FSDA, $\{D_k^A\}$ and $\{L_{1,k}^A K_k^A (L_{2,k}^A)^\top\}$ converge to zero, $\{D_k^H\}$ and $\{L_k^H K_k^H (L_k^H)^\top\}$ converge to D^X and L_{lr}^X and, $\{D_k^G\}$ and $\{L_k^G K_k^G (L_k^G)^\top\}$ converge to D^Y and L_{lr}^Y , respectively, all quadratically.

Proof. It follows from (26) that $\{A_k\}$ converges to zero. Then the decomposition $A_k = D_k^A + L_{1,k}^A K_k^A (L_{2,k}^A)^\top$ in (8) together with $\lim_{k \rightarrow \infty} D_k^A = 0$ imply that the sequence $\{L_{1,k}^A K_k^A (L_{2,k}^A)^\top\}$ will converge to zero quadratically.

Additionally, as the sequences $\{H_k\}$ and $\{G_k\}$ converge quadratically, by (26), to the unique solutions X_s and Y_s , respectively, and

$$H_k = D_k^H + L_k^H K_k^H (L_k^H)^\top, \quad G_k = D_k^G + L_k^G K_k^G (L_k^G)^\top$$

in (8). So, given the initial banded matrices $D_0^H = D^H$ and $D_0^G = D^G$, the iterations $\{D_k^H\}$ and $\{D_k^G\}$ in (9) and (13) are independent of the low-ranked part and have the unique limits D^X and D^Y , respectively. Consequently, the sequences $\{L_k^H K_k^H (L_k^H)^\top\} = \{H_k - D_k^H\}$ and $\{L_k^G K_k^G (L_k^G)^\top\} = \{G_k - D_k^G\}$ converge quadratically to the matrices $X_s - D^X := L_{lr}^X$ and $Y_s - D^Y := L_{lr}^Y$, respectively. \square

Remark 2. 1. Although the product $L_{1,k}^A K_k^A (L_{2,k}^A)^\top$ converges to zero, it follows from (15), (17) and (23) that the kernel K_k^A and low-rank factors $L_{1,k}^A$ and $L_{2,k}^A$ might still not converge to zero, respectively.

2. If the convergence of SDA (or the corresponding FSDA) is quadratic, the number of the iterations k is not big when termination occurs, then the matrices L_{lr}^X and L_{lr}^Y are generally of numerical low-rank.

To show the evolution of the bandwidth of D_k^A , D_k^G and D_k^H , we first require the following result [37].

Theorem 2. Let $A = (a_{ij})$ be an $n \times n$ matrix. Assume that there is a number m such that $a_{ij} = 0$ if $|i - j| > m$ and that $\|A\| \leq c_1$ and $\|A^{-1}\| \leq c_2$ for some $c_1 > 0$ and $c_2 > 0$. Then for $A^{-1} = (\alpha_{ij})$, there are numbers $K > 0$ and $0 < r < 1$ depending only on c_1, c_2 and m , such that

$$|\alpha_{ij}| \leq Kr^{|i-j|} \text{ for all } i, j.$$

We now consider the evolution of the bandwidth for the banded parts.

Theorem 3. Let $b_k^a = bw(D_k^A)$, $b_k^g := bw(D_k^G)$ and $b_k^h := bw(D_k^H)$ for $k \geq 0$. If the assumption (25) holds, then for iteration scheme (13), there is an integers \bar{k} independent of k , such that

$$\begin{aligned} b_k^a &\leq 2^{\bar{k}} b_0^a + (2^{\bar{k}} - 1) \log_r^{(\tau/K)}, \\ b_k^g &\leq (2^{\bar{k}+1} - 2) b_0^g + b_0^g + (2^{\bar{k}+1} - 2 - \bar{k}) \log_r^{(\tau/K)}, \\ b_k^h &\leq (2^{\bar{k}+1} - 2) b_0^h + b_0^h + (2^{\bar{k}+1} - 2 - \bar{k}) \log_r^{(\tau/K)}, \end{aligned}$$

where τ is the truncation tolerance and $K > 0$ and $0 < r < 1$ depend only on the upper bounds of $\|I + D_i^H D_i^G\|$, $\|I + D_i^G D_i^H\|$, $\|(I + D_i^H D_i^G)^{-1}\|$ and $\|(I + D_i^G D_i^H)^{-1}\|$ for $i \leq \bar{k}$.

Proof. It follows from [35] (Thm 4.6) that $I - D_k^H D_k^G$ and $I - D_k^G D_k^H$ are non-singular for all k . This together with (29) indicate that there is an integer \bar{k} such that $|(D_{\bar{k}}^A)_{ij}| < \tau$ and the increment of D_k^G and D_k^H in (13) satisfies

$$|(D_{\bar{k}}^A (I + D_{\bar{k}}^G D_{\bar{k}}^H)^{-1} D_{\bar{k}}^G D_{\bar{k}}^{A\top})_{ij}| < \tau \text{ and } |(D_{\bar{k}}^{A\top} (I + D_{\bar{k}}^H D_{\bar{k}}^G)^{-1} D_{\bar{k}}^H D_{\bar{k}}^A)_{ij}| < \tau, \quad (30)$$

where τ is the given the truncation tolerance. On the other hand for $k = 1, \dots, \bar{k}$, it follows from Theorem 2 that there are $K > 0$ and $0 < r < 1$ independent of k , such that

$$|((I + D_k^G D_k^H)^{-1})_{ij}| \leq Kr^{|i-j|}, \quad |((I + D_k^H D_k^G)^{-1})_{ij}| \leq Kr^{|i-j|}.$$

Then one has

$$bw((I + D_k^G D_k^H)^{-1}) \leq \log_r^{(\tau/K)}, \quad bw((I + D_k^H D_k^G)^{-1}) \leq \log_r^{(\tau/K)}$$

for $k \leq \bar{k}$. Now recalling the iteration (9), the bandwidths of the first iteration admit the bounds

$$b_1^a \leq 2b_0^a + \log_r^{(\tau/K)}, \quad b_1^s \leq 2b_0^a + b_0^s + \log_r^{(\tau/K)}, \quad b_1^h \leq 2b_0^a + b_0^h + \log_r^{(\tau/K)}.$$

Iterating the above bandwidth bounds according to the scheme (13) at $k \geq 1$, we have

$$\begin{aligned} b_k^a &\leq 2^k b_0^a + (2^k - 1) \log_r^{(\tau/K)}, \\ b_k^s &\leq (2^{k+1} - 2) b_0^a + b_0^s + (2^{k+1} - 2 - k) \log_r^{(\tau/K)}, \\ b_k^h &\leq (2^{k+1} - 2) b_0^a + b_0^h + (2^{k+1} - 2 - k) \log_r^{(\tau/K)}. \end{aligned} \quad (31)$$

In particular, the bounds on the RHS of (31) will attain the maximal values at $k = \bar{k}$ since elements with the absolute value less than τ are removed as in (30). \square

3. Deflation of Low-Rank Factors and Kernels

It has been shown that there is an exponential increase in the dimension of low-rank factors and kernels. Nevertheless, it is clear that the first three items in $L_{1,k}^A$ and $L_{2,k}^A$ (see (15) and (17)) are same as the second to the fourth item in L_k^G and L_k^H (see (14) and (16)), respectively. Then the deflation of low-rank factors and kernels is needed to keep these matrices low-ranked. To see this process clearly, we start with the case $k = 2$.

Case for $k = 2$.

Consider the deflation of the low-rank factors firstly. It follows from (14)–(17) that

$$\begin{aligned} L_2^G &= [L_1^G, L_{11}^A, D_1^{AGH} L_1^G, D_1^{AGHG} L_1^H, D_1^{AGHG} L_{21}^A], \\ L_{12}^A &= [L_{11}^A, D_1^{AGH} L_1^G, D_1^{AGHG} L_1^H, D_1^{AGH} L_{11}^A], \\ L_2^H &= [L_1^H, L_{21}^A, D_1^{A\top HG} L_1^H, D_1^{A\top HGH} L_1^G, D_1^{A\top HGH} L_{11}^A], \\ L_{22}^A &= [L_{21}^A, D_1^{A\top HG} L_1^H, D_1^{A\top HGH} L_1^G, D_1^{A\top HG} L_{21}^A] \end{aligned}$$

with

$$\begin{aligned} D_1^{AGHG} &= D_1^A (I + D_1^G D_1^H)^{-1} D_1^G, & D_1^{A\top HGH} &= (D_1^A)^\top D_1^H (I + D_1^G D_1^H)^{-1}, \\ D_1^{AGH} &= D_1^A (I + D_1^G D_1^H)^{-1}, & D_1^{A\top HG} &= (D_1^A)^\top (I + D_1^H D_1^G)^{-1}. \end{aligned}$$

Expanding the above low-rank factors with the initial $L_{10}^A \in \mathbb{R}^{N \times m^a}$ and $L_{20}^A \in \mathbb{R}^{N \times m^a}$, one can see from Appendix A that L_{10}^A and $D_1^{AGHG} L_{20}^A$ (or L_{20}^A and $D_1^{A\top HGH} L_{10}^A$) occur twice in L_2^G (or L_2^H). To reduce the dimension of L_2^G , we remove the duplicated L_{10}^A in L_1^G (or L_{20}^A in L_1^H) and retain the one in L_{11}^A (or L_{21}^A). Furthermore, we remove $D_1^{AGHG} L_{20}^A$ in $D_1^{AGHG} L_{21}^A$ (or $D_1^{A\top HGH} L_{10}^A$ in $D_1^{A\top HGH} L_{11}^A$) and keep the one in $D_1^{AGHG} L_1^H$ (or $D_1^{A\top HGH} L_1^G$). Then the

original L_2^G (or L_2^H) is deflated to L_2^{Gd} (or L_2^{Hd}) of a smaller dimension, where the superscript “d” indicates the matrix after deflation. Analogously, as $D_1^{AGH}L_{10}^A$ and $D_1^{A^T HG}L_{20}^A$ appear twice in L_{12}^A and L_{22}^A , we apply the same deflation process to L_{12}^A and L_{22}^A , respectively, obtaining L_{12}^{Ad} and L_{22}^{Ad} in Appendix A, where the left blank in each factor corresponds to the deleted matrix and the black bold matrices inherit from the undeflated ones. Note that the deflated matrices L_2^{Gd} , L_{12}^{Ad} , L_2^{Hd} and L_{22}^{Ad} are still denoted by L_2^G , L_{12}^A , L_2^H and L_{22}^A , respectively, in next iteration to simplify notations.

For the kernels at $k = 2$, one has

$$K_2^G = \begin{bmatrix} 2m^a & 4m^a & 2m^a & 2m^a \\ K_1^G & 0 & 0 & 0 \\ 0 & K_1^{AGHGA^T} & K_1^{AGHG} & K_1^A \\ 0 & (K_1^{AGHG})^T & -K_1^{GHG} & 0 \\ 0 & (K_1^A)^T & 0 & 0 \end{bmatrix} \begin{matrix} 2m^a \\ 4m^a \\ 2m^a \\ 2m^a \end{matrix},$$

$$K_2^H = \begin{bmatrix} 2m^a & 4m^a & 2m^a & 2m^a \\ K_1^H & 0 & 0 & 0 \\ 0 & K_1^{A^T HGHA} & K_1^{A^T HGH} & (K_1^A)^T \\ 0 & (K_1^{A^T HGH})^T & -K_1^{HGH} & 0 \\ 0 & K_1^A & 0 & 0 \end{bmatrix} \begin{matrix} 2m^a \\ 4m^a \\ 2m^a \\ 2m^a \end{matrix}$$

and

$$K_2^A = \begin{bmatrix} 2m^a & 4m^a & 2m^a \\ K_1^{AGHA} & K_1^{AGH} & K_1^A \\ (K_1^{A^T GH})^T & -K_1^{GH} & 0 \\ K_1^A & 0 & 0 \end{bmatrix} \begin{matrix} 2m^a \\ 4m^a \\ 2m^a \end{matrix},$$

with non-zero components defined in (18)–(20). Here, details of the deflation of K_2^G are explained explicitly and that for K_2^H is similar. In fact, there are 10 block rows and block columns with each of initial size $m^a \times m^a$ in K_2^G . Due to the deflation of the L -factors described above, we add the first and the ninth row to the third and the seventh row and then remove the first and the ninth row, respectively. We also add the the first and the ninth column to the third and the seventh column and then remove the first and the ninth column, respectively, completing the deflation of K_2^{Gd} .

Analogously, there are eight block rows and block columns, each of the initial size $m^a \times m^a$ in K_2^A . The deflation process simultaneously adds the seventh column and row subblocks to the third column and row subblocks, respectively. Then the first column sub-block of the upper right K_1^A and the first row sub-block of the lower-left K_1^A overlap with the first column sub-block of K_1^{AGH} and the first row sub-block of $(K_1^{A^T GH})^T$, respectively, completing the deflation of K_2^{Ad} .

The whole process is described in Figures 1 and 2 where each small square is of size $m^a \times m^a$ and each block with gray background represents the non-zero component in K_2^G and K_2^A . The little white squares in K_2^{Gd} and K_2^{Ad} inherit from the originally undeflated submatrices and the little black squares in K_2^{Gd} and K_2^{Ad} represent the submatrices after summation.

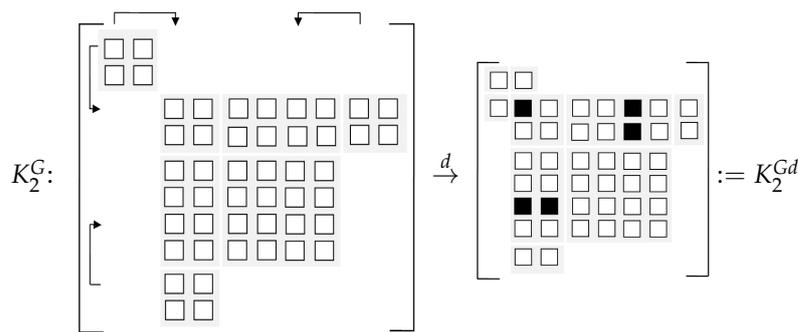


Figure 1. The deflation process of K_2^G (or K_2^H).

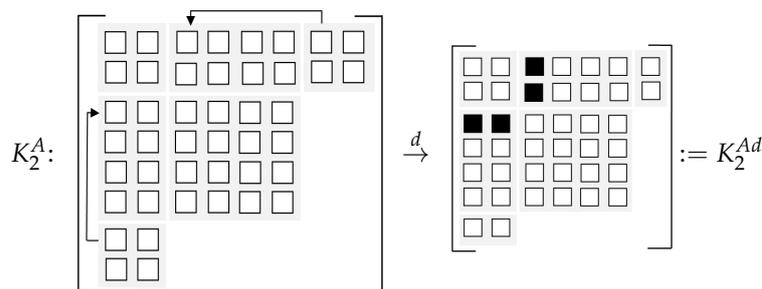


Figure 2. The deflation process of K_2^A .

Case for $k \geq 3$.

After the $(k - 1)$ -th deflation, the deflated matrices L_{k-1}^{Gd} , $L_{1,k-1}^{Ad}$, L_{k-1}^{Hd} and $L_{2,k-1}^{Ad}$ are denoted by L_{k-1}^G , $L_{1,k-1}^A$, L_{k-1}^H and $L_{2,k-1}^A$ for simplicity. Now there are $m_{k-1}^s - (k - 1)m^a$ (or $m_{k-1}^h - (k - 1)m^a$) columns in L_{k-1}^G and $L_{1,k-1}^A$ (or L_{k-1}^H and $L_{2,k-1}^A$) and $m_{k-1}^{a2} - m^a$ (or $m_{k-1}^{a1} - m^a$) columns in $D_{k-1}^{AGHG}L_{2,k-1}^A$ and $D_{k-1}^{AGHG}L_{k-1}^H$ (or $D_{k-1}^{A^T HGH}L_{1,k-1}^A$ and $D_{k-1}^{A^T HGH}L_{k-1}^G$) that are identical. Then, one can remove columns of

$$L_{k-1}^G(:, (k - 2)m^a + 1 : m_{k-1}^s - m^a) \text{ (or } L_{k-1}^H(:, (k - 2)m^a + 1 : m_{k-1}^h - m^a) \text{)}$$

and

$$D_{k-1}^{AGHG}L_{2,k-1}^A(:, 1 : m_{k-1}^{a2} - m^a) \text{ (or } D_{k-1}^{A^T HGH}L_{1,k-1}^A(:, 1 : m_{k-1}^{a1} - m^a) \text{)},$$

and keep the columns of

$$L_{1,k-1}^A(:, 1 : m_{k-1}^s - (k - 1)m^a) \text{ (or } L_{2,k-1}^A(:, 1 : m_{k-1}^h - (k - 1)m^a) \text{)}$$

and

$$D_{k-1}^{AGHG}L_{k-1}^H(:, m_{k-1}^h - m_{k-1}^{a2} + 1 : m_{k-1}^h - m^a) \text{ (or } D_{k-1}^{A^T HGH}L_{k-1}^G(:, m_{k-1}^s - m_{k-1}^{a1} + 1 : m_{k-1}^s - m^a) \text{)}$$

in L_k^G (A1) (or L_k^H (A3)), respectively. So there are $k - 1$ matrices, each of order $N \times m^a$, that are left in L_{k-1}^G (or L_{k-1}^H), i.e., $D_0^{AGHG}L_{20}^A$, $D_1^{AGHG}D_0^{A^T HGH}L_{20}^A$, ..., $D_{k-2}^{AGHG}\Pi_{i=0}^{k-3}D_i^{A^T HGH}L_{20}^A$ in (A1) (or $D_0^{A^T HGH}L_{10}^A$, $D_1^{A^T HGH}D_0^{AGHG}L_{10}^A$, ..., $D_{k-2}^{A^T HGH}\Pi_{i=0}^{k-3}D_i^{AGHG}L_{10}^A$ in (A3)) in Appendix B. Meanwhile, only one matrix of order $N \times m^a$ is left in $D_{k-1}^{AGHG}L_{2,k-1}^A$ (or $D_{k-1}^{A^T HGH}L_{1,k-1}^A$), i.e., the last item $D_{k-1}^{AGHG}\Pi_{i=k-2}^0D_i^{A^T HGH}L_{20}^A$ in (A1) (or $D_{k-1}^{A^T HGH}\Pi_{i=k-2}^0D_i^{AGHG}L_{10}^A$ in (A3)) of Appendix B. We also take L_3^G as an example to describe the above deflation more clearly in Appendix C.

To deflate $L_{1,k}^A$ ($L_{2,k}^A$), columns of

$$D_{k-1}^{AGH} L_{1,k-1}^A(:, 1 : m_{k-1}^{a_1} - m^a) \text{ (or } D_{k-1}^{A^\top HG} L_{2,k-1}^A(:, 1 : m_{k-1}^{a_2} - m^a) \text{)}$$

are removed but the columns of

$$D_{k-1}^{AGH} L_{k-1}^G(:, m_{k-1}^g - m_{k-1}^{a_1} + 1 : m_{k-1}^g - m^a) \text{ (or } D_{k-1}^{A^\top HG} L_{k-1}^H(:, m_{k-1}^h - m_{k-1}^{a_2} + 1 : m_{k-1}^h - m^a) \text{)}$$

are retained in $L_{1,k}^A$ (or $L_{2,k}^A$). So only one matrix of order $N \times m^a$ is left in $D_{k-1}^{AGH} L_{1,k-1}^A$ (or $D_{k-1}^{A^\top HG} L_{2,k-1}^A$), i.e., the last item $\Pi_{i=k-1}^0 D_i^{AGH} L_{10}^A$ in (A2) (or $\Pi_{i=k-1}^0 D_i^{A^\top HG} L_{20}^A$ in (A4)) of Appendix B. Note that the low-rank factors in the $(k - 1)$ -th iteration are the ones after deflation, truncation and compression, deleting the superscript “d” for the simplicity. We take L_{13}^A as an example to describe the above deflation more clearly in Appendix D.

Correspondingly, the kernel matrices K_k^G , K_k^H , and K_k^A are deflated according to their low-rank factors. Here, we describe the deflation of K_k^G and that of K_k^H is essentially the same. By recalling the place of non-zero sub-matrices (the block with gray background in Figure 3) of K_k^G in (21), the deflation process essentially adds $K_{k-1}^G((k - 2)m^a + 1 : m_{k-1}^g - m^a, (k - 2)m^a + 1 : m_{k-1}^g - m^a)$ to $K_{k-1}^{AGHGA^\top}(1 : m_{k-1}^g - (k - 1)m^a, 1 : m_{k-1}^g - (k - 1)m^a)$, columns $K_{k-1}^A(:, 1 : m_{k-1}^{a_2} - m^a)$ to $K_{k-1}^{AGHG}(:, m_{k-1}^g + m_{k-1}^h - m_{k-1}^{a_2} + 1 : m_{k-1}^g + m_{k-1}^h - m^a)$ and rows $(K_{k-1}^A)^\top(1 : m_{k-1}^{a_2} - m^a, :)$ to $(K_{k-1}^{AGHG})^\top(m_{k-1}^g + m_{k-1}^h - m_{k-1}^{a_2} + 1 : m_{k-1}^g + m_{k-1}^h - m^a, :)$, respectively. See Figure 3 for illustration.

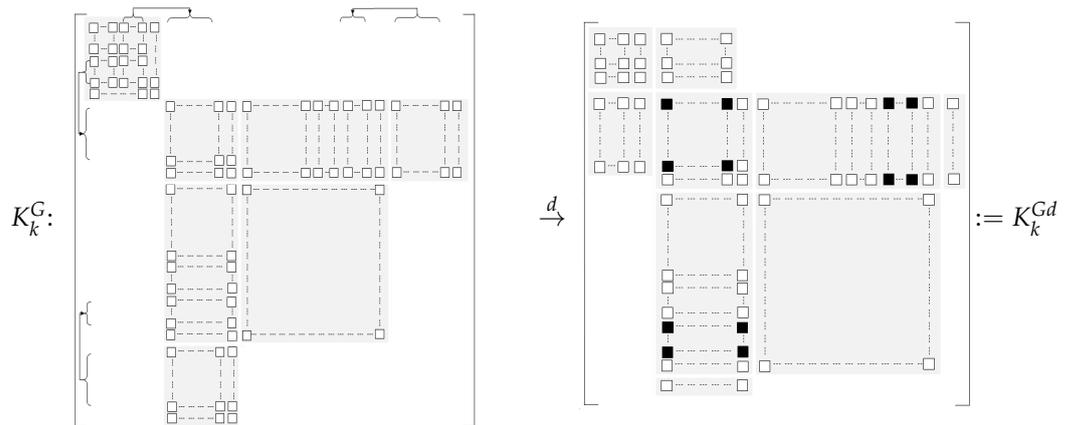


Figure 3. The deflation process of K_k^G (or K_k^H).

Similarly, by recalling the positions of non-zero matrices (the block with gray background in Figure 4) of K_k^A in (23), the deflation process will add columns $K_{k-1}^A(:, 1 : m_{k-1}^{a_2} - m^a)$ to columns $K_{k-1}^{AHG}(:, m_{k-1}^h - m_{k-1}^{a_2} + 1 : m_{k-1}^h - m^a)$ and rows $K_{k-1}^A(1 : m_{k-1}^{a_1} - m^a, :)$ to rows $(K_{k-1}^{AGH})^\top(m_{k-1}^g - m_{k-1}^{a_1} + 1 : m_{k-1}^g - m^a, :)$. See Figure 4 for illustration.

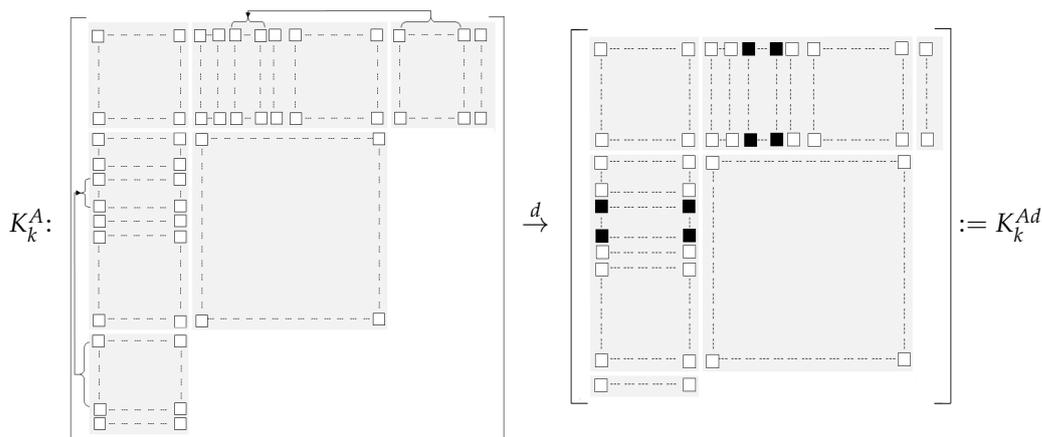


Figure 4. The deflation process of K_k^A .

4. Partial Truncation and Compression

Although the deflation of the low-rank factors and kernels in the last section can reduce dimensional growth, the exponential increment of the undeflated part is still rapid, making large-scale computation and storage infeasible. Conventionally, one efficient way to shrink the column number of low-rank factors is by truncation and compression (TC) [17,18], which, unfortunately, is hard to be applied to our case due to the following two main obstacles.

- Direct application of TC to L_k^{Hd} , L_k^{Gd} , $L_{1,k}^{Ad}$, $L_{2,k}^{Ad}$, and their corresponding kernels K_k^{Hd} , K_k^{Gd} and K_k^{Ad} at the k -th step will require four QR decompositions, resulting in a relatively high computational complexity and CPU consumption.
- The TC process applied to the whole low-rank factors at current step breaks up the implicit structure, causing the deflation to be unrealized in the next iteration.

In this section, we will instead present a technique of partial truncation and compression (PTC) to overcome the above difficulties. Our PTC only requires two QR decompositions of the exponentially increasing (not the entire) parts of low-rank factors, keeping the successive deflation for subsequent iterations.

PTC for low-rank factors. Recall the deflated forms (A1) and (A3) in Appendix B. L_k^{Gd} and L_k^{Hd} can be divided to three parts

$$\begin{aligned} L_k^{Gd} &= [L_k^{Gd}(1), L_k^{Gd}(2), L_k^{Gd}(3)] \\ L_k^{Hd} &= [L_k^{Hd}(1), L_k^{Hd}(2), L_k^{Hd}(3)]. \end{aligned}$$

The number of columns in

$$L_k^{Gd}(1) := [D_0^{AGHG} L_{20}^A, D_1^{AGHG} D_0^{A^\top GH} L_{20}^A, \dots, D_{k-2}^{AGHG} \Pi_{i=k-3}^0 D_i^{A^\top GH} L_{20}^A] \in \mathbb{R}^{N \times (k-1)m^a}$$

and

$$L_k^{Hd}(1) := [D_0^{A^\top HGH} L_{10}^A, D_1^{A^\top HGH} D_0^{A^\top GH} L_{10}^A, \dots, D_{k-2}^{A^\top HGH} \Pi_{i=k-3}^0 D_i^{AGH} L_{10}^A] \in \mathbb{R}^{N \times (k-1)m^a}$$

increases only linearly with k , and the last parts

$$L_k^{Gd}(3) := D_{k-1}^{AGHG} \Pi_{i=k-2}^0 D_i^{A^\top GH} L_{20}^A \in \mathbb{R}^{N \times m^a}$$

and

$$L_k^{Hd}(3) := D_{k-1}^{A^\top HGH} \Pi_{i=k-2}^0 D_i^{AGH} L_{10}^A \in \mathbb{R}^{N \times m^a}$$

are always of size $N \times m^a$. So we only truncate and compress the dominantly growing parts

$$L_k^{Gd}(2) := [L_{1,k-1}^A, D_{k-1}^{AGH} L_{k-1}^G, D_{k-1}^{AGHG} L_{k-1}^H]$$

and

$$L_k^{Hd}(2) := [L_{2,k-1}^A, D^{A^\top HG} L_{k-1}^H, D_{k-1}^{A^\top HGH} L_{k-1}^G]$$

by orthogonalization. Consider the QR decompositions with column pivoting of

$$\begin{aligned} L_k^{Gd}(2)P_k^G &= [Q_k^G \tilde{Q}_k^G] \begin{bmatrix} U_{k,1}^G & U_{k,2}^G \\ 0 & \tilde{U}_k^G \end{bmatrix}, \quad \|\tilde{U}_k^G\| < u_0^g \tau_g, \\ L_k^{Hd}(2)P_k^H &= [Q_k^H \tilde{Q}_k^H] \begin{bmatrix} U_{k,1}^H & U_{k,2}^H \\ 0 & \tilde{U}_k^H \end{bmatrix}, \quad \|\tilde{U}_k^H\| < u_0^h \tau_h, \end{aligned} \tag{32}$$

where P_k^G and P_k^H are permutation matrices such that the diagonal elements of $\begin{bmatrix} U_{k,1}^J & U_{k,2}^J \\ 0 & \tilde{U}_k^J \end{bmatrix}$ ($J = G$ or H) are decreasing in absolute value, $u_0^g = \|U_{0,1}^G\|$, $u_0^h = \|U_{0,1}^H\|$ and τ_g and τ_h are some small tolerances controlling PTC of $L_k^{Gd}(2)$ and $L_k^{Hd}(2)$, respectively, $m_k^{g(2)}$ and $m_k^{h(2)}$ are the respective column numbers of $L_k^G(2)$ and $L_k^H(2)$ bounded above by some given m_{\max} . Then their ranks satisfy

$$r_k^g := \text{rank}(L_k^G(2)) \leq m_k^{g(2)} \leq m_{\max}, \quad r_k^h := \text{rank}(L_k^H(2)) \leq m_k^{h(2)} \leq m_{\max}$$

with $m_{\max} \ll N$. Furthermore, $Q_k^G \in \mathbb{R}^{N \times r_k^g}$ and $Q_k^H \in \mathbb{R}^{N \times r_k^h}$ are orthonormal and $U_k^G = [U_{k,1}^G \ U_{k,2}^G] \in \mathbb{R}^{r_k^g \times m_{k-1}^{hga}}$ and $U_k^H = [U_{k,1}^H \ U_{k,2}^H] \in \mathbb{R}^{r_k^h \times m_{k-1}^{hga}}$ are full-rank with $m_{k-1}^{hga} = m_{k-1}^h + m_{k-1}^g + m_{k-1}^a$. Then L_k^{Gd} and L_k^{Hd} can be truncated and reorganized as

$$\begin{aligned} L_k^{Gdt} &= [L_k^{Gd}(1), Q_k^G, L_k^{Gd}(3)] := [L_k^{Gdt}(1), L_k^{Gdt}(2), L_k^{Gdt}(3)] \in \mathbb{R}^{N \times m_k^g}, \\ L_k^{Hdt} &= [L_k^{Hd}(1), Q_k^H, L_k^{Hd}(3)] := [L_k^{Hdt}(1), L_k^{Hdt}(2), L_k^{Hdt}(3)] \in \mathbb{R}^{N \times m_k^h} \end{aligned} \tag{33}$$

with $m_k^g = r_k^g + km^a$ and $m_k^h = r_k^h + km^a$.

Similarly, recalling the deflated forms in (A2) and (A4) in Appendix B, $L_{1,k}^{Ad}$ and $L_{2,k}^{Ad}$ are also divided into two parts,

$$L_{1,k}^{Ad} = [L_{1,k}^{Ad}(1), L_{1,k}^{Ad}(2)] \quad \text{and} \quad L_{2,k}^{Ad} = [L_{2,k}^{Ad}(1), L_{2,k}^{Ad}(2)]$$

with

$$\begin{aligned} L_{1,k}^{Ad}(1) &= L_k^{Gd}(2), \quad L_{1,k}^{Ad}(2) = \Pi_{i=k-1}^0 D_i^{AGH} L_{10}^A, \\ L_{2,k}^{Ad}(1) &= L_k^{Hd}(2), \quad L_{2,k}^{Ad}(2) = \Pi_{i=k-1}^0 D_i^{A^\top HG} L_{20}^A. \end{aligned}$$

Since $L_k^{Gd}(2)$ and $L_k^{Hd}(2)$ have been compressed to Q_k^G and Q_k^H , respectively, one has the truncated and compressed factors

$$\begin{aligned} L_{1,k}^{Adt} &= [Q_k^G, L_{1,k}^{Ad}(2)] = [L_{1,k}^{Adt}(1), L_{1,k}^{Adt}(2)] \in \mathbb{R}^{N \times m_k^{a1}}, \\ L_{2,k}^{Adt} &= [Q_k^H, L_{2,k}^{Ad}(2)] = [L_{2,k}^{Adt}(1), L_{2,k}^{Adt}(2)] \in \mathbb{R}^{N \times m_k^{a2}} \end{aligned} \tag{34}$$

with $m_k^{a1} = r_k^g + m^a$ and $m_k^{a2} = r_k^h + m^a$, finishing the PTC process for the low-rank factors in the k -th iteration.

It is worth noting that the above PTC process can proceed to the next iteration. In fact, one has

$$\begin{aligned} L_{k+1}^G &= [L_k^{Gdt}, L_{1,k}^{Adt}, D_k^{AGH} L_k^{Gdt}, D_k^{AGHG} L_k^{Hdt}, D_{k+1}^{AGHG} L_{2,k}^{Adt}], \\ L_{k+1}^H &= [L_k^{Hdt}, L_{2,k}^{Adt}, D_k^{A^\top HG} L_k^{Hdt}, D_k^{A^\top HGH} L_k^{Gdt}, D_k^{A^\top HGH} L_{1,k}^{Adt}] \end{aligned}$$

after the k -th PTC. As $L_{1,k}^{Adt}(1)$ is equal to $L_k^{Gdt}(2)$ and $L_{2,k}^{Adt}(1)$ is equal to $L_k^{Hdt}(2)$, one can deflate L_{k+1}^G and L_{k+1}^H to

$$L_{k+1}^{Gd} = [L_{k+1}^{Gd}(1), L_{k+1}^{Gd}(2), L_{k+1}^{Gd}(3)], \quad L_{k+1}^{Hd} = [L_{k+1}^{Hd}(1), L_{k+1}^{Hd}(2), L_{k+1}^{Hd}(3)]$$

with

$$\begin{aligned} L_{k+1}^{Gd}(1) &= [L_k^{Gdt}(1), L_k^{Gdt}(3)], \quad L_{k+1}^{Hd}(1) = [L_k^{Hdt}(1), L_k^{Hdt}(3)], \\ L_{k+1}^{Gd}(2) &= [L_{1,k}^{Adt}, D_k^{AGH}L_k^{Gdt}, D_k^{AGHG}L_k^{Hdt}], \quad L_{k+1}^{Hd}(2) = [L_{2,k}^{Adt}, D_k^{A^{\top}HG}L_k^{Hdt}, D_k^{A^{\top}HGH}L_k^{Gdt}], \\ L_{k+1}^{Gd}(3) &= D_k^{AGHG}L_{2,k}^{Adt}(2), \quad L_{k+1}^{Hd}(3) = D_k^{A^{\top}HGH}L_{1,k}^{Adt}(2). \end{aligned}$$

Applying PTC to $L_{k+1}^{Gd}(2)$ and $L_{k+1}^{Hd}(2)$, respectively, again, one has

$$\begin{aligned} L_{k+1}^{Gdt} &= [L_{k+1}^{Gd}(1), Q_{k+1}^G L_{k+1}^{Gd}(3)] := [L_{k+1}^{Gdt}(1), L_{k+1}^{Gdt}(2), L_{k+1}^{Gdt}(3)], \\ L_{k+1}^{Hdt} &= [L_{k+1}^{Hd}(1), Q_{k+1}^H L_{k+1}^{Hd}(3)] := [L_{k+1}^{Hdt}(1), L_{k+1}^{Hdt}(2), L_{k+1}^{Hdt}(3)], \end{aligned} \tag{35}$$

where $Q_{k+1}^G \in \mathbb{R}^{N \times r_{k+1}^g}$ and $Q_{k+1}^H \in \mathbb{R}^{N \times r_{k+1}^h}$ are unitary matrices from QR decomposition and the PTC in the $(k + 1)$ -th iteration is completed.

PTC for kernels. Define matrices

$$\begin{aligned} \hat{U}_{1,k}^A &= U_k^G \oplus I_{m^a}, \quad \hat{U}_k^G = I_{(k-1)m^a} \oplus U_k^G \oplus I_{m^a}, \\ \hat{U}_{2,k}^A &= U_k^H \oplus I_{m^a}, \quad \hat{U}_k^H = I_{(k-1)m^a} \oplus U_k^H \oplus I_{m^a}, \end{aligned}$$

with U_k^G and U_k^H in (32). Then the truncated and compressed kernels are

$$\begin{aligned} K_k^{Gdt} &:= \hat{U}_k^G K_k^{Gd} (\hat{U}_k^G)^{\top} \in \mathbb{R}^{m_k^g \times m_k^g}, \\ K_k^{Hdt} &:= \hat{U}_k^H K_k^{Hd} (\hat{U}_k^H)^{\top} \in \mathbb{R}^{m_k^h \times m_k^h}, \\ K_k^{Adt} &:= \hat{U}_{1,k}^A K_k^{Ad} (\hat{U}_{2,k}^A)^{\top} \in \mathbb{R}^{m_k^g \times m_k^h}. \end{aligned} \tag{36}$$

To eliminate items less than $O(\tau_g)$ and $O(\tau_h)$ in the low-rank factors and kernels, an additional monitoring step is imposed after the PTC process. Specifically, the last item $D_{k-2}^{AGHG} \Pi_{i=k-3}^0 D_i^{A^{\top}GH} L_{20}^A$ in L_k^{Gdt} (or $D_{k-2}^{A^{\top}HGH} \Pi_{i=k-3}^0 D_i^{AGH} L_{10}^A$ in L_k^{Hdt}) will be discarded if its norm is less than $O(\tau_g)$ (or $O(\tau_h)$). Similarly, $\Pi_{i=k-1}^0 D_i^{AGH} L_{10}^A$ in $L_{1,k}^{Ad}(2)$ (or $\Pi_{i=k-1}^0 D_i^{A^{\top}HG} L_{20}^A$ in $L_{2,k}^{Ad}(2)$) will be abandoned if its norm is less than $O(\tau_g)$ (or $O(\tau_h)$). In this way, the growth of column dimension in the low-rank factors L_k^{Gdt} , L_k^{Hdt} , $L_{1,k}^{Adt}$ and $L_{2,k}^{Adt}$, as well as the kernels K_k^{Gdt} , K_k^{Hdt} , K_k^{Adt} , will be controlled efficiently while sacrificing a hopefully negligible bit of accuracy. Additionally, their sizes after PTC will be further restricted by setting a reasonable upper bound m_{\max} .

5. Algorithm and Implementation

5.1. Computation of Residuals

The computation of relative residuals, such as $r_{rel} = |\mathcal{D}(H_k)|/|\mathcal{D}(H_0)|$, is commonly used in the context of solving the DARE using SDA, as mentioned in [4]. Typically, the FSDA algorithm is designed to stop when the relative residual is sufficiently small, which guarantees that the approximated solution H_k is close to the exact solution of the DARE [35]. However, computing r_{rel} directly can be computationally expensive due to the high rank of H_k and G_k . To overcome this difficulty, the residual is divided into two parts, the banded part and the low-ranked part, under the assumptions of Equations (4) and (5). The residual for the banded part can be computed relatively easily and serves as a pre-termination condition, followed by the termination of the entire FSDA algorithm based on the residual for the low-ranked part.

5.1.1. Residual for the Banded Part

Define

$$\tilde{D}_k^{HG} = (I + D_k^H D_0^G)^{-1}, \quad \tilde{D}_k^{HGH} = \tilde{D}_k^{HG} D_k^H, \quad \tilde{D}_k^{GHG} = D_0^G \tilde{D}_k^{HG}$$

and

$$\tilde{K}_k^H = (I + K_k^H (L_k^H)^\top \tilde{D}_k^{GHG} L_k^H)^{-1} K_k^H.$$

With the current approximated solution $H_k = D_k^H + L_k^H K_k^H (L_k^H)^\top$, the residual for DARE (3) is

$$\begin{aligned} \mathcal{D}(H_k) &= -H_k + A^\top \left(\tilde{D}_k^{HGH} + \tilde{D}_k^{HG} L_k^H \tilde{K}_k^H (\tilde{D}_k^{HG} L_k^H)^\top \right) A + H \\ &:= D_k^R + L_k^R K_k^R (L_k^R)^\top, \end{aligned}$$

where the banded part, the low-rank part and the kernel are

$$\begin{aligned} D_k^R &= D_0^H - D_k^H + (D_0^A)^\top D_k^H (I + D_0^G D_k^H)^{-1} D_0^A, \\ L_k^R &= [L_{20}^A, (D_0^A)^\top \tilde{D}_k^{HGH} L_{10}^A, (D_0^A)^\top \tilde{D}_k^{HG} L_k^H, L_k^H], \\ K_k^R &= \begin{bmatrix} m^a & m^a & m_k^h & m_k^h \\ \tilde{K}_k^{A^\top HGHA} & I_{m^a} & \tilde{K}_k^{A^\top HG} & 0 \\ I_{m^a} & 0 & 0 & 0 \\ (\tilde{K}_k^{A^\top HG})^\top & 0 & \tilde{K}_k^H & 0 \\ 0 & 0 & 0 & -K_k^H \end{bmatrix} \begin{bmatrix} m^a \\ m^a \\ m_k^h \\ m_k^h \end{bmatrix} \end{aligned} \tag{37}$$

respectively, and

$$\begin{aligned} \tilde{K}_k^{A^\top HG} &= (L_{10}^A)^\top \tilde{D}_k^{HG} L_k^H \cdot \tilde{K}_k^H, \\ \tilde{K}_k^{A^\top HGHA} &= (L_{10}^A)^\top \tilde{D}_k^{HGH} L_{10}^A + \tilde{K}_k^{A^\top HG} \cdot \left((L_{10}^A)^\top \tilde{D}_k^{HG} L_k^H \right)^\top. \end{aligned}$$

It is not difficult to see that the main flop counts in the kernel K_k^R lie in forming matrices

$$(L_{10}^A)^\top \tilde{D}_k^{HGH} L_{10}^A, \quad (L_{10}^A)^\top \tilde{D}_k^{HG} L_k^H, \quad (L_k^H)^\top \tilde{D}_k^{GHG} L_k^H. \tag{38}$$

To avoid calculating them in each iteration, we first verify if

$$\text{B_RRes} = \frac{\|D_k^R\|}{|\bar{D}_0^R| + \|L_0^R\|^2 \|K_0^R\|} \leq \epsilon_b \tag{39}$$

with $|\bar{D}_0^R| = \|D_0^A\|_2^2 \|D_0^H\| \|(I + D_0^G D_0^H)^{-1}\|_2$ and ϵ_b being the band tolerance. Here, the norm $\|\cdot\|_2$ is the matrix spectral norm, which is not easy to compute and is replaced by l_1 -matrix norm in practice. This is feasible as the residual of $\mathcal{D}(H_k)$ comes from two relatively independent parts, i.e., the banded part and the low-rank part.

5.1.2. Residual for the Low-Rank Part

When the pre-termination (39) is satisfied, matrices in (38) are then constructed, followed by the deflation, truncation, and compression of the low-rank factor L_k^R . Specifically, the columns $L_{20}^A(:, 1 : m^a)$ are removed and columns of $L_k^H(:, 1 : m^a)$ are kept such that L_k^R is deflated to L_k^{Rd} , i.e.,

$$\begin{aligned}
 L_k^R &= \left[\overbrace{L_{20}^A, (D_0^A)^\top \tilde{D}_k^{HGH} L_{10}^A, (D_0^A)^\top \tilde{D}^{HG} L_k^H, L_k^H}^{1 : m^a \rightarrow m_k^h + 2m^a + 1 : m_k^h + 3m^a} \right] \\
 &\xrightarrow{d} \left[(D_0^A)^\top \tilde{D}_k^{HGH} L_{10}^A, (D_0^A)^\top \tilde{D}^{HG} L_k^H, L_k^H \right] \\
 &:= L_k^{Rd}.
 \end{aligned}$$

Let $\hat{I}_{m^a} = [I_{m^a}, 0, \dots, 0] \in \mathbb{R}^{m^a \times m_k^h}$, $\hat{K}_k^{A^\top HG} = [(\tilde{K}_k^{A^\top HG})^\top, 0, \dots, 0] \in \mathbb{R}^{m_k^h \times m_k^h}$. The kernel K_k^R in (37) is correspondingly deflated as

$$K_k^R \xrightarrow{d} \begin{bmatrix} m^a & m_k^h & m_k^h \\ 0 & 0 & \hat{I}_{m^a} \\ 0 & \tilde{K}_k^H & \hat{K}_k^{A^\top HG} \\ (\hat{I}_{m^a})^\top & (\hat{K}_k^{A^\top HG})^\top & \hat{K}_k^{A^\top HGHA} \end{bmatrix} \begin{matrix} m^a \\ m_k^h \\ m_k^h \end{matrix} := K_k^{Rd},$$

where all elements in $\hat{K}_k^{A^\top HGHA}$ are same to those in K_k^H except $\hat{K}_k^{A^\top HGHA}(1 : m^a, 1 : m^a) = \tilde{K}_k^{A^\top HGHA} - K_k^H(1 : m^a, 1 : m^a)$.

After deflation, the truncation and compression are applied to L_k^{Rd} with QR decomposition

$$L_k^{Rd} P_k^R = [Q_k^R \tilde{Q}_k^R] \begin{bmatrix} U_{k,1}^R & U_{k,2}^R \\ 0 & \tilde{U}_k^R \end{bmatrix}, \quad \|\tilde{U}_k^R\| < u_0^r \tau_r,$$

where P_k^R is the permutation matrix such that the diagonal elements of $\begin{bmatrix} U_{k,1}^R & U_{k,2}^R \\ 0 & \tilde{U}_k^R \end{bmatrix}$ are decreasing in absolute value, $u_0^r = \|U_{0,1}^R\|$ and τ_r is the given tolerance, $Q_k^R \in \mathbb{R}^{n \times r_k^r}$ is orthonormal and $U_k^R = [U_{k,1}^R \ U_{k,2}^R] \in \mathbb{R}^{r_k^r \times n_k}$ is full-ranked. Since $\|L_k^R K_k^R (L_k^R)^\top\| \approx \|U_k^R K_k^{Rd} (U_k^R)^\top\|$, the terminating condition of the whole algorithm is chosen to be

$$LR_RRes = \frac{\|U_k^R K_k^{Rd} (U_k^R)^\top\|}{|\tilde{D}_0^R| + \|L_0^R\|^2 \|K_0^R\|} \leq \epsilon_l \tag{40}$$

with ϵ_l being the low-rank tolerance.

5.2. Algorithm and Operation Counts

The process of deflation and PTC together with the computation of residuals (39) and (40) are summarized in the FSDA Algorithm 1.

Algorithm 1 FSDA. Solve DAREs with high-ranked G and H

Inputs: Banded matrices D_0^A, D_0^G, D_0^H , low-rank factors $L_{10}^A, L_{20}^A, L_0^G, L_0^H, K_0^G, K_0^H$, and the iterative tolerance tol ; truncation tolerances τ_g, τ_h, τ_r and upper bound m_{max} ; band tolerance ϵ_b and low-rank tolerance ϵ_l .

Outputs: Banded matrix D^H , low-rank matrix L^H and the kernel matrix K^H with the stabilizing solution $X_s \approx D^H + L^H K^H (L^H)^\top$.

1. Set $D_1^G = D_0^G + D_0^{AGHG} (D_0^A)^\top, D_1^H = D_0^H + D_0^{A^\top HGH} D_0^A, D_1^A = D_0^A (D_0^{A^\top HG})^\top$ as in (9). Set $L_1^G = [L_{10}^A, D_0^{AGHG} L_{20}^A], L_1^H = [L_{20}^A, D_0^{A^\top HGH} L_{10}^A], L_{11}^A = [L_{10}^A, D_0^{AGH} L_{10}^A], L_{21}^A = [L_{20}^A, D_0^{A^\top HG} L_{20}^A]$ as in (10). Set K_1^G, K_1^H, K_1^A as in (11) and (12).
2. **For** $k = 2, \dots$, until convergence, **do**
3. Compute banded matrices D_k^G, D_k^H, D_k^A as in (13).
4. Form components (18)–(20) and construct kernels K_k^G, K_k^H and K_k^A as in (21)–(23).
5. Deflate kernels $K_k^G \xrightarrow{d} K_k^{Gd}, K_k^H \xrightarrow{d} K_k^{Hd}$ and $K_k^A \xrightarrow{d} K_k^{Ad}$ in a way of Figures 3 and 4.
6. Deflate the low-rank factors $L_k^G \xrightarrow{d} L_k^{Gd}, L_k^H \xrightarrow{d} L_k^{Hd}, L_{1,k}^A \xrightarrow{d} L_{1,k}^{Ad}$ and $L_{2,k}^A \xrightarrow{d} L_{2,k}^{Ad}$ as in (A1)–(A4).
7. Partially truncate and compress L_k^{Gd} and L_k^{Hd} as in (32) with accuracy $u_0^g \tau_g, u_0^h \tau_h$.
8. Construct compressed low-rank factors $L_k^{Gdt}, L_k^{Hdt}, L_{1,k}^{Adt}$ and $L_{2,k}^{Adt}$ as in (33)–(34).
9. Construct compressed kernels K_k^{Gdt}, K_k^{Hdt} and K_k^{Adt} as in (36).
10. Evaluate the residual of the banded part B_RRes in (39).
11. **If** B_RRes $< tol$, compute the residual of low-rank part LR_RRes in (40).
12. **If** LR_RRes $< tol$, break, end.
13. **End (If);**
14. $K_k^G := K_k^{Gdt}, K_k^H := K_k^{Hdt}, K_k^A := K_k^{Adt}$.
15. $L_k^G := L_k^{Gdt}, L_k^H := L_k^{Hdt}, L_{1,k}^A := L_{1,k}^{Adt}, L_{2,k}^A := L_{2,k}^{Adt}$.
16. $k := k + 1$;
17. **End (For)**
18. Output $D_k^H = D^H, L_k^H = L^H$ and $K_k^H = K^H$.

Remark 3. 1. At each iteration, elements in the banded matrices D_k^A, D_k^H , and D_k^G with an absolute value less than $tol = eps \cdot \max\{\|D^A\|, \|D^G\|, \|D^H\|\}$ are eliminated.

2. The deflation process involves merging selected rows and columns in the kernels K_k^G, K_k^H , and K_k^A based on overlapping columns in the low-rank factors $L_k^G, L_k^H, L_{1,k}^A$, and $L_{2,k}^A$. This requires adding some columns and rows.

3. The PTC is applied to $L_k^{Gd}(2)$ and $L_k^{Hd}(2)$. The column numbers of $L_k^{Gd}(1)$ and $L_k^{Hd}(1)$ increase linearly with respect to k , while those of $L_k^{Gd}(3)$ and $L_k^{Hd}(3)$ remain unchanged. Elements in $L_k^{Gd}(1), L_k^{Hd}(1), L_k^{Gd}(3)$, and $L_k^{Hd}(3)$ with an absolute value less than tol are removed to minimize the column size of the low-rank factors.

To further analyze the complexity and the memory requirement of the FSDA, the bandwidth of D_k^A, D_k^G , and D_k^H at each iteration are assumed to be b_k^a, b_k^g and $b_k^h (b_k^a, b_k^g, b_k^h \ll N)$, respectively. We also set $b_k^{hg} = \max\{b_k^h, b_k^g\}, b_k^{hga} = \max\{b_k^h, b_k^g, b_k^a\}, m_k^a = \max\{m_k^{a1}, m_k^{a2}\}$, and $m_{k-1}^{hga} := m_{k-1}^h + m_{k-1}^g + m_{k-1}^a$ for the convenience of counting flops. The table in Appendix E lists the time and memory requirement for different components in the k -th iteration of the FSDA, where the estimations are upper bounds due to the truncation errors τ_g, τ_h and τ_r .

6. Numerical Examples

In this section, we will demonstrate the effectiveness of the FSDA algorithm in computing the approximate solution of the DARE (3). The FSDA algorithm was implemented using MATLAB 2014a [38] on a 64-bit PC running Windows 10. The PC had a 3.0 GHz Intel

Core i5 processor with 6 cores and 6 threads, 32GB RAM, and a machine unit round-off value of $\text{eps} = 2.22 \times 10^{-16}$. The residual for the DARE was estimated using the upper bound formula

$$\tilde{r}_k = \text{B_RRes} + \text{LR_RRes},$$

where B_RRes in (39) and LR_RRes in (40) are the relative residuals for the banded part and the low-rank part, respectively. The tolerance values for truncation and compression were set to $\tau_g = \tau_h = \tau_r = 10^{-16}$, and the termination tolerance values were set to $\epsilon_b = \epsilon_l = 10^{-11}$. We also tried $N \cdot \text{eps}$ as the tolerance value for τ_g , τ_h and τ_r in our experiments, but found that it had no impact on the residual accuracy. The maximum permitted column number in the low-rank factors was set to $m_{\max} = 2200$. As a comparison, we also ran the ordinary SDA algorithm with hierarchical structure (i.e., HODLR) using the hm-toolbox (<http://github.com/numpi/hm-toolbox>, accessed on 1 June 2023) [39,40]. The SDA algorithm with hierarchical structure is referred to as SDA_HODLR in this paper. The derived relative residual for SDA_HODLR is denoted by \hat{r}_k . In our numerical experiments, the initial bandwidths of all banded matrices in Examples 1 and 3 were relatively small, while those in Example 2 were non-trivial.

Example 1. *The first example is of the medium scale, measuring the error between the true solution and the computed one. Given the constant $\theta = \sqrt{\eta + \frac{1}{\eta} - 2\zeta}$, where ζ and η are positive numbers such that θ is real. Let $L_{10}^A = \theta e$ with e the random vector satisfying $e^\top e = 1$, $L_{20}^A = L_{10}^A$, $D_0^A = \zeta I_N$, then $A = D_0^A + L_{10}^A (L_{20}^A)^\top$. Set $G = D_0^G = I_N$, $H = D_0^H = (\eta + \frac{1}{\eta}) D_0^A - (D_0^A)^2 - I_N$. The solution of the DARE is of the form $X_s = D^X + L^X (L^X)^\top$ with $D^X = \eta D_0^A - I_N$ and $L^X = \sqrt{\eta} L_{10}^A$.*

It is not difficult to see that the solution X_s is stabilizing since the spectral radius of $(I_N + GX_s)^{-1}A$ is less than unity when $\eta > 1$.

We first took $\zeta = 1.2$ and $\eta = 2$ to calculate B_RRes, followed by LR_RRes as well as the upper bound of residual of DARE \tilde{r}_k . In our implementations, the relative error between the approximated solution (denoted by H_j when terminated at the j -th iteration) and the true stabilizing solution X_s was evaluated, and the numerical results are presented in Table 1. It is seen that for different scales ($N = 1000, 3000, 5000, 7000$) FSDA was able to attain the prescribed banded accuracy in five iterations. Residuals LR_Res and \tilde{r}_k were then evaluated, attaining the order $O(10^{-16})$. The relative error with the computational time being not included in the CPU time, also reflects that H_5 approximates the true solution very well. On the other hand, SDA_HODLR also attains the prescribed residual accuracy in five iterations, but cost more CPU time (in seconds).

We then took $\eta = 1.2$ to make the spectral radius of $(I_N + GX_s)^{-1}A$ close to 1 and recorded the numerical performance of the FSDA with $\zeta = 1.0$. It is seen from Table 1 that the FSDA costs seven iterations before termination, obtaining almost the same banded residual histories (B_RRes) for different N . As before, LR_RRes and \tilde{r}_k were of $O(10^{-17})$ and $O(10^{-16})$, respectively, showing that H_7 is a good approximation to the true solution to DARE (3). The last relative error $\|H_7 - X_s\|/\|X_s\|$ also validates this fact. Analogously, SDA_HODLR requires seven iterations to arrive at the residual level $O(10^{-15})$. It is also seen that the FSDA costs less CPU time than SDA_HODLR for all N .

Table 1. Residual and actual errors in Example 1.

$\zeta = 1.2, \eta = 2.0$				
N	1000	3000	5000	7000
FSDA				
B_RRes	4.39×10^{-1}	4.41×10^{-1}	4.42×10^{-1}	4.42×10^{-1}
	3.47×10^{-2}	3.48×10^{-2}	3.49×10^{-2}	3.49×10^{-2}
	1.38×10^{-4}	1.38×10^{-4}	1.38×10^{-4}	1.38×10^{-4}
	2.10×10^{-9}	2.11×10^{-9}	2.11×10^{-9}	2.11×10^{-9}
	4.25×10^{-16}	4.27×10^{-16}	4.27×10^{-16}	4.31×10^{-16}
LR_RRes	2.09×10^{-18}	2.27×10^{-18}	4.04×10^{-18}	3.28×10^{-18}
\tilde{r}_k	4.27×10^{-16}	4.29×10^{-16}	4.31×10^{-16}	4.34×10^{-16}
$\ H_5 - X_s\ /\ X_s\ $	2.56×10^{-16}	2.57×10^{-16}	2.56×10^{-16}	2.48×10^{-16}
CPU	0.04	0.09	0.22	0.48
SDA_HODLR				
\hat{r}_k	4.44×10^{-1}	4.44×10^{-1}	4.44×10^{-1}	4.44×10^{-1}
	3.50×10^{-2}	3.50×10^{-2}	3.50×10^{-2}	3.50×10^{-2}
	1.39×10^{-4}	1.39×10^{-4}	1.39×10^{-4}	1.39×10^{-4}
	2.12×10^{-9}	2.12×10^{-9}	2.12×10^{-9}	2.12×10^{-9}
	1.33×10^{-15}	1.27×10^{-15}	1.34×10^{-15}	1.47×10^{-15}
CPU	1.17	19.93	76.67	186.61
$\zeta = 1.0, \eta = 1.2$				
N	1000	3000	5000	7000
FSDA				
B_RRes	8.68×10^{-1}	8.84×10^{-1}	8.89×10^{-1}	8.92×10^{-1}
	6.06×10^{-1}	6.18×10^{-1}	6.21×10^{-1}	6.23×10^{-1}
	1.93×10^{-1}	1.97×10^{-1}	1.98×10^{-1}	1.99×10^{-1}
	1.15×10^{-2}	1.18×10^{-2}	1.18×10^{-2}	1.19×10^{-2}
	3.40×10^{-5}	3.47×10^{-5}	3.49×10^{-5}	3.50×10^{-5}
	2.91×10^{-10}	2.97×10^{-10}	2.99×10^{-10}	3.00×10^{-10}
	8.22×10^{-16}	8.38×10^{-16}	8.43×10^{-16}	8.46×10^{-16}
LR_RRes	3.03×10^{-17}	1.07×10^{-17}	2.77×10^{-17}	1.75×10^{-17}
\tilde{r}_k	8.52×10^{-16}	8.48×10^{-16}	8.70×10^{-16}	8.63×10^{-16}
$\ H_7 - X_s\ /\ X_s\ $	4.23×10^{-15}	5.04×10^{-15}	4.94×10^{-15}	4.98×10^{-15}
CPU	0.31	0.45	0.48	0.96
SDA_HODLR				
\hat{r}_k	9.08×10^{-1}	9.08×10^{-1}	9.08×10^{-1}	9.08×10^{-1}
	6.34×10^{-1}	6.34×10^{-1}	6.34×10^{-1}	6.34×10^{-1}
	2.02×10^{-1}	2.02×10^{-1}	2.02×10^{-1}	2.02×10^{-1}
	1.21×10^{-2}	1.21×10^{-2}	1.21×10^{-2}	1.21×10^{-2}
	3.56×10^{-5}	3.56×10^{-5}	3.56×10^{-5}	3.56×10^{-5}
	3.05×10^{-10}	3.05×10^{-10}	3.05×10^{-10}	3.05×10^{-10}
	4.75×10^{-15}	4.62×10^{-15}	4.97×10^{-15}	5.52×10^{-15}
CPU	1.61	27.10	107.16	263.34

Example 2. Consider a generalized model of power system labelled by PI Sections 20–80 (<https://sites.google.com/site/rommes/software>, “S10PI_n1.mat” accessed on 1 June 2023). All transmission lines in the network are modelled by RLC ladder networks, of cascaded RLC PI-circuits [41]. The original band-plus-low-rank matrix A has a small scale of 528 (Figure 5) and is then extended

to larger ones. Specifically, we extract the banded part D_{ori}^A of the bandwidth 217 from the original matrix A_{ori} and tile it along the diagonal direction for 20 times to obtain D_0^A . We then implement an SVD of the matrix $A_{\text{ori}} - D_{\text{ori}}^A$ to produce the singular value matrix Σ_A and the unitary matrices U_A and V_A . The low-ranked parts L_{10}^A and L_{20}^A are then constructed by tiling $U_A(:, 1 : r_a)$ and $V_A(:, 1 : r_a)$ 20 times and multiplying $\Sigma_A^{1/2}(1 : r_a, 1 : r_a)$ from the right, respectively, where r_a is the number of singular values in Σ_A less than 10^{-8} . Let F_1 and F_3 be block diagonal matrices with each diagonal block the 3×3 random matrix (generated by 'rand(3)'). Let F_2 and F_4 be also diagonal block matrices with the top left element a random number, the last diagonal block 2×2 random matrix and others 3×3 random matrices. Define matrices G and H as

$$G := D_0^G = (R_g + R_g^T)/2 + \zeta I_N, \quad H := D_0^H = (R_h + R_h^T)/4 + \zeta I_N,$$

with $R_g = (F_1 + I_N)(F_2 + I_N)$, $R_h = (F_3 + I_N)(F_4 + I_N)$.

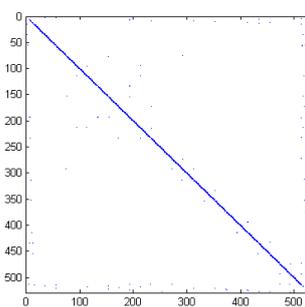


Figure 5. Structured matrix A_{ori} of size 528×528 in Example 2.

We ran the FSDA with three different $\zeta = 0.11, 1.0, 3.0$, each conducting five random experiments. In all experiments, B_RRes and LR_RRes (in log 10) were observed attaining the pre-terminating condition (39) and the terminating condition (40), respectively.

Figure 6 plots the obtained numerical results for five experiments, where Rk is the upper bound of the residual of the DARE, BRes and LRes are the absolute residuals of the banded part and the low-rank part (i.e., the numerators in B_RRes and LR_RRes), respectively. It is seen that the relative residual levels of LR_RRes and B_RRes (between 10^{-14} and 10^{-17}) are lower than those of LRes and BRes (between 10^{-11} and 10^{-13}) in all experiments. Particularly, the gap between them increases as ζ becomes larger. On the other hand, the residual line of Rk is above the residual lines of B_RRes or LR_RRes, attaining the level between 10^{-15} and 10^{-16} . This demonstrates that the FSDA can obtain a relatively high residual accuracy.

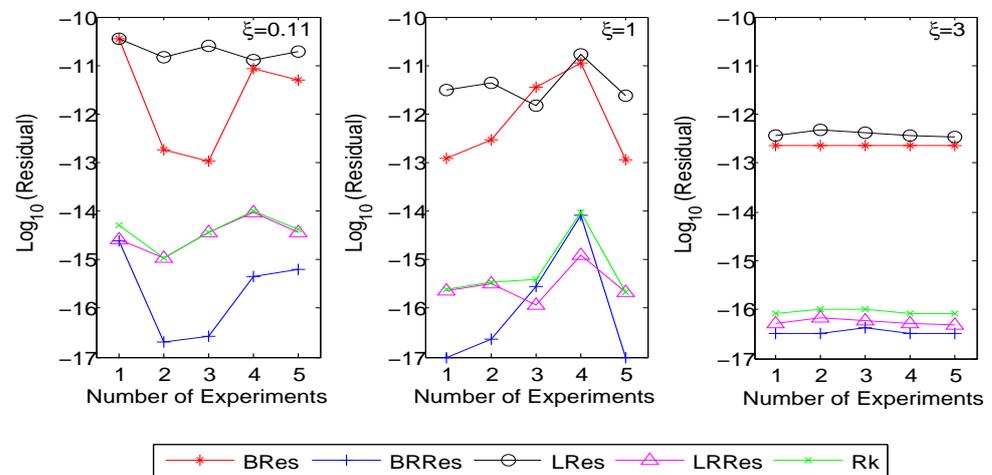


Figure 6. Residual of the banded part and the low-rank part for different ζ .

To clearly see the evolution of the bandwidth of the banded matrices and the dimensional increase in the low-rank factors for five iterations, we listed the history of bandwidths of D_k^G , D_k^H , and D_k^A (denoted by b_k^g , b_k^h , and b_k^a , respectively) and the column numbers of L_k^{Hdt} and L_k^{Gdt} (denoted by m_k^h and m_k^g , respectively) in Table 2, where the CPU row recorded the consumed CPU time in seconds. It is obviously seen that, for $\zeta = 0.11, 1$, and 3 , the FSDA requires 5, 4, and 3 iterations to reach the prescribed accuracy, respectively. Further experiments show that the required number of iterations, when terminated, will decrease as ζ goes larger. Additionally, we see that bandwidths b_k^g and b_k^h rise much in the second iteration but keep almost unchanged for the remaining iterations. Nevertheless, b_k^a decreases gradually after reaching the maximal value in the second iteration, which is consistent with the convergence of D_k^A in Corollary 1. On the other hand, we see from m_k^h and m_k^g that the column numbers in the second iteration are about fourfold of those in the first iteration since the FSDA does not deflate the low-rank factors at the first iteration. However, the column numbers in the fifth iteration (if it exists) are less than twofold of those in the fourth iteration. This reflects that deflation and PTC are efficient in reducing the dimensions of low-rank factors. In our experiments, we also found that nearly half of the CPU time in the FSDA was consumed in forming $(I_N + D_k^H D_0^G)^{-1} D_k^H$ in the pre-termination. However, such a time expense might decrease if the initial bandwidths b_0^g , b_0^h , and b_0^a are narrow.

Table 2. CPU times and history of bandwidth of banded matrices and column numbers of low-rank factors in Example 2.

	1	2	3	4	5
	$[b_k^g \ b_k^h \ b_k^a \ m_k^h \ m_k^g]$	$[b_k^g \ b_k^h \ b_k^a \ m_k^h \ m_k^g]$	$[b_k^g \ b_k^h \ b_k^a \ m_k^h \ m_k^g]$	$[b_k^g \ b_k^h \ b_k^a \ m_k^h \ m_k^g]$	$[b_k^g \ b_k^h \ b_k^a \ m_k^h \ m_k^g]$
$\zeta = 0.11$	[445 445 445 34 34] [979 980 981 126 132] [981 980 980 474 484] [981 980 768 1012 1020] [981 980 519 1758 1767]	[445 445 445 34 34] [982 982 1042 126 132] [981 980 980 474 481] [981 980 768 1014 1018] [981 980 522 1759 1771]	[445 445 445 34 34] [973 767 973 126 132] [973 767 748 480 492] [973 767 674 1025 1032] [973 767 493 1801 1812]	[445 445 445 34 34] [1047 1033 1051 126 132] [1050 1047 1049 468 495] [1050 1042 1047 1096 1028] [1050 1042 983 1946 1853]	[445 445 445 34 34] [998 998 997 126 132] [998 999 973 474 488] [981 980 768 1011 1023] [981 980 525 1762 1773]
CPU	4443.63	4451.36	4456.96	4414.65	4457.14
$\zeta = 1$	[445 445 445 34 34] [973 767 973 126 132] [973 973 973 471 476] [973 973 646 911 927]	[445 445 445 34 34] [768 973 769 126 132] [767 973 766 469 476] [767 973 555 910 916]	[445 445 445 34 34] [815 973 973 126 132] [815 973 768 477 487] [815 973 646 1007 1027]	[445 445 445 34 34] [1033 996 1042 126 132] [1042 1042 1042 479 490] [1042 1042 840 973 980]	[445 445 445 34 34] [745 745 748 126 132] [753 980 732 474 488] [753 980 684 923 931]
CPU	4014.65	4025.74	3993.86	4107.84	4020.12
$\zeta = 3.0$	[445 445 445 34 34] [652 654 674 126 132] [652 654 519 448 453]	[445 445 445 34 34] [746 746 746 126 132] [746 746 650 466 475]	[445 445 445 34 34] [695 673 675 126 132] [695 673 614 449 454]	[445 445 445 34 34] [674 686 685 126 132] [674 686 658 447 454]	[445 445 445 34 34] [701 703 686 126 132] [701 703 651 448 455]
CPU	1797.39	1640.02	1803.23	1748.16	1695.01

To further compare numerical performances between the FSDA and SDA_HODLR for larger problems, we extended the original scale to $N = 15,840, 21,120, 26,400$ and $31,680$ at $\zeta = 3.0$ and ran both algorithms until convergence. The results are listed in Table 3, where one can see that both the FSDA and SDA_HODLR (i.e., SDA_HD in the table) attain the prescribed residual accuracy within three iterations, and SDA_HODLR requires less CPU time than FSDA does. However, there seems a strong tendency that the FSDA will outperform the SDA_HODLR on CPU time for larger problems, as the CPU time of the SDA_HODLR appears to surge at $N = 26,400$ and SDA_HODLR used up memory at $N = 31,680$ without producing any numerical results (denoted by “—”). The symbols “*” in the SDA_HODLR column represent no related records for bandwidth and column number of the low-rank factors.

We further modified this example to have a simpler banded part to test both algorithms. Specifically, the relatively data-concentrated banded part of bandwidth 3 is extracted and tiled along the diagonal direction for 20 times to form D_0^A . As before, an SVD is imposed on the rest matrix to construct the low-ranked parts L_{10}^A and L_{20}^A after tiling the derived unitary

matrices 20 times and multiplying $\Sigma_A^{1/2}(1 : r_a, 1 : r_a)$ from the right. We still selected $\zeta = 3.0$ and ran both the FSDA and SDA_HODLR at scales $N = 15,840, 21,120, 26,400$ and $31,680$ again. The obtained results are recorded in Table 4, where it is readily seen that the FSDA outperforms the SDA_HODLR on CPU time. Once again, the SDA_HODLR ran out of memory for the case $N = 31,680$.

Table 3. Numerical results for FSDA and SDA_HODLR in Example 2 at $\zeta = 3.0$. The symbol * stands for no related records.

N	15,840		21,120		26,400		31,680	
	FSDA	SDA_HD	FSDA	SDA_HD	FSDA	SDA_HD	FSDA	SDA_HD
b_k^g	[445 695 695]	*	[445 736 736]	*	[445 723 723]	*	[445 652 652]	*
b_k^h	[445 673 673]	*	[445 745 745]	*	[445 737 737]	*	[445 654 654]	*
b_k^a	[445 675 614]	*	[445 745 674]	*	[445 738 653]	*	[445 674 619]	*
m_k^h	[34 126 448]	*	[34 126 469]	*	[34 126 460]	*	[34 126 444]	*
m_k^g	[34 132 453]	*	[34 132 476]	*	[34 132 469]	*	[34 132 454]	*
IT.	3	3	3	3	3	3	3	—
RES.	7.83×10^{-17}	1.44×10^{-15}	7.27×10^{-17}	1.70×10^{-15}	8.04×10^{-17}	1.74×10^{-15}	5.96×10^{-15}	—
CPU	6740.54	1285.31	13,037.43	3701.43	18,154.14	17,653.63	21,618.03	—

Table 4. Numerical results for FSDA and SDA_HODLR in relatively simpler banded part of Example 2 at $\zeta = 3.0$. The symbol * stands for no related records.

N	15,840		21,120		26,400		31,680	
	FSDA	SDA_HD	FSDA	SDA_HD	FSDA	SDA_HD	FSDA	SDA_HD
b_k^g	[31 31 31]	*	[36 37 37]	*	[38 39 39]	*	[34 37 37]	*
b_k^h	[28 30 30]	*	[34 36 36]	*	[38 40 40]	*	[36 39 39]	*
b_k^a	[28 31 28]	*	[36 38 34]	*	[38 42 38]	*	[34 40 35]	*
m_k^h	[48 280 628]	*	[48 286 647]	*	[48 287 651]	*	[48 285 647]	*
m_k^g	[48 281 628]	*	[48 285 645]	*	[48 287 650]	*	[48 287 650]	*
IT.	3	3	3	3	3	3	3	—
RES.	5.95×10^{-17}	2.09×10^{-15}	3.46×10^{-16}	1.81×10^{-15}	8.49×10^{-16}	3.05×10^{-15}	9.93×10^{-17}	—
CPU	133.71	1255.06	218.07	3744.06	288.53	15,508.81	344.18	—

Example 3. This example is an extension of small-scale electric power systems networks to a large-scale one which is used for signal stability analysis [19–21]. The corresponding matrix A_{ori} is from the power system of New England (<https://sites.google.com/site/rommes/software>, “*ww_36_pemc_36.mat*”, accessed on 1 June 2023). Figure 7 presents the original structure of the matrix A of order 66. We properly modified elements $A_{ori}(32, 28) = -36.4687$, $A_{ori}(32, 29) = -37.922$, $A_{ori}(46, 42) = -33.0033$; $A_{ori}(46, 43) = -76.8277$, $A_{ori}(60, 56) = -83.0405$, $A_{ori}(60, 57) = -73.9947$, $A_{ori}(60, 59) = -34.0478$. Then the banded part D_{ori}^A is extracted from blocks $A_{ori}(1:6, 1:6)$, $A_{ori}(7:13, 7:13)$, $A_{ori}(14:20, 14:20)$, $A_{ori}(21:27, 21:27)$, $A_{ori}(28:34, 28:34)$, $A_{ori}(35:41, 35:41)$, $A_{ori}(42:48, 42:48)$, $A_{ori}(49:55, 49:55)$, $A_{ori}(56:62, 56:62)$, and $A_{ori}(63:66, 63:66)$, admitting the bandwidth of 4. After tiling D_{ori}^A 200, 400, and 600 times along the diagonal direction, we obtain banded matrix D_0^A of scales $N = 13,200, 26,400$ and $39,600$. For the low-rank factors, an SVD of the matrix $A_{ori} - D_{ori}^A$ is firstly implemented to produce the diagonal singular value matrix Σ_A and the unitary matrices U_A and V_A . The low-ranked parts L_{10}^A and L_{20}^A are then constructed by tiling $U_A(:, 1 : r_a)$ and $V_A(:, 1 : r_a)$ 200, 400, and 600 times and dividing their F -norms, respectively, where r_a is the number of singular values in Σ_A less than 10^{-10} . The matrices G and H are

$$G := D_0^G = \zeta I_N, \quad H := D_0^H = I_N - \frac{1}{1 + \zeta} D_0^A (D_0^A)^T$$

with $\zeta > 0$.

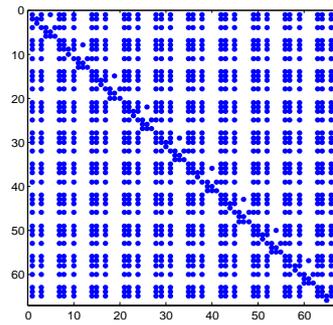


Figure 7. Structured matrix A of order 66×66 (1194 non-zeros) in Example 3.

We took different ζ and ran the FSDA to compute the stabilizing solution for different dimensions $N = 13,200, 26,400,$ and $39,600$. In our experiments, the FSDA always satisfied the pre-terminating condition (39) first and then terminated at $LR_RRes < \epsilon_l = 10^{-11}$. We picked $\zeta = 95$ and listed derived results in Table 5, where BRes (or LRes) and B_RRes (or LR_RRes) record the absolute and the relative residual for the banded part (or the low-rank part), respectively, and $\tilde{r}_k, [b_k^g, b_k^h, b_k^a, m_k^h, m_k^g]$ record histories of the upper bound of the residual of DARE, the bandwidths of D_k^G, D_k^H and D_k^A and the column numbers of the low-rank factors L_k^{Hdt} and L_k^{Gdt} , respectively. Particularly, the t_k column describes the accumulated time to compute residuals (excluding the data marked with “*”).

Table 5. Residuals, column numbers of low-rank factors, and CPU times at $\zeta = 95$ in Example 3.

k	BRes	B_RRes	LRes	LR_RRes	\tilde{r}_k	$[b_k^g, b_k^h, b_k^a, m_k^h, m_k^g]$	t_k
$N = 13,200 \quad \zeta = 95 \quad \tau_g = \tau_h = 10^{-16} \quad m_{max} = 2000$							
1	1.02×10^3	1.42×10^{-2}	1.04×10^3 *	1.25×10^{-2} *	2.67×10^{-2} *	[6 6 6 29 29]	1.03
2	2.33×10^0	3.25×10^{-5}	1.40×10^{-1} *	2.06×10^{-6} *	3.45×10^{-5} *	[6 6 6 66 66]	5.01
3	6.19×10^{-3}	8.64×10^{-8}	2.94×10^{-3} *	4.33×10^{-8} *	1.30×10^{-7} *	[6 6 6 76 76]	79.13
4	1.37×10^{-7}	2.02×10^{-12}	6.28×10^{-6}	7.76×10^{-11}	7.96×10^{-11}	[6 6 6 100 101]	158.63
5	2.27×10^{-9}	3.30×10^{-14}	4.31×10^{-11}	6.33×10^{-16}	3.38×10^{-14}	[6 6 6 169 170]	246.55
$N = 26,400 \quad \zeta = 95 \quad \tau_g = \tau_h = 10^{-16} \quad m_{max} = 2000$							
1	8.31×10^2	8.64×10^{-3}	1.61×10^3 *	1.81×10^{-2} *	2.67×10^{-2} *	[6 6 6 29 29]	3.58
2	2.95×10^0	3.07×10^{-5}	1.40×10^{-1} *	1.46×10^{-5} *	3.21×10^{-5} *	[6 6 6 66 66]	13.92
3	4.91×10^{-3}	5.11×10^{-8}	2.94×10^{-3} *	3.06×10^{-8} *	8.07×10^{-8} *	[6 6 6 75 76]	534.56
4	1.94×10^{-7}	2.02×10^{-12}	5.28×10^{-6}	5.49×10^{-11}	5.69×10^{-11}	[6 6 6 97 98]	1085.76
5	3.21×10^{-9}	3.30×10^{-14}	4.81×10^{-11}	8.00×10^{-16}	3.39×10^{-14}	[6 6 6 160 161]	1675.01
$N = 39,600 \quad \zeta = 95 \quad \tau_g = \tau_h = 10^{-16} \quad m_{max} = 2000$							
1	1.01×10^3	8.64×10^{-3}	1.62×10^3 *	1.81×10^{-2} *	2.67×10^{-2} *	[6 6 6 29 29]	7.93
2	3.61×10^0	3.07×10^{-5}	1.40×10^{-1} *	1.19×10^{-6} *	3.19×10^{-5} *	[6 6 6 66 66]	33.41
3	6.02×10^{-3}	5.11×10^{-8}	2.94×10^{-3} *	2.50×10^{-8} *	7.62×10^{-8} *	[6 6 6 76 77]	605.64
4	2.37×10^{-7}	2.02×10^{-12}	5.28×10^{-6}	4.48×10^{-11}	4.68×10^{-11}	[6 6 6 100 102]	1210.54
5	3.94×10^{-9}	3.30×10^{-14}	5.22×10^{-11}	4.43×10^{-16}	3.39×10^{-14}	[6 6 6 170 172]	1923.38

Obviously, for different N , the FSDA is capable of achieving the prescribed accuracy after five iterations. The residuals BRes, B_RRes, LRes, and LR_RRes indicate that the FSDA tended to converge quadratically. Especially, BRes (or B_RRes) at different N are of nearly same order and terminate at $O(10^{-9})$ (or $O(10^{-11})$). Similarly, LRes (or LR_RRes) at different N attain the order $O(10^{-11})$ (or $O(10^{-16})$). More iterations seemed useless in improving the accuracy of LRes and LR_RRes. Note that data labelled with the superscript “*” in columns LRes, LR_RRes and \tilde{r}_k come from the re-running of the FSDA to complement the residual in each iteration, and their corresponding CPU time is not included in the

column t_k . Lastly, $[b_k^g b_k^h b_k^a m_k^h m_k^g]$ indicate that the bandwidths of $D_k^G, D_k^H,$ and D_k^A are invariant and the column numbers of the low-rank factors grow less than twice in each iteration, demonstrating the effectiveness of the deflation and PTC.

We also ran the FSDA to compute the solution of the DARE of $\zeta = 90$ and the results were recorded in Table 6. In this case, the FSDA requires seven iterations to reach the prescribed accuracy. As before, the last few residuals in the column BRes (or B_RRes) at different N are almost the same of $O(10^{-9})$ (or $O(10^{-14})$). The residuals LRes (or LR_RRes) at different N terminate at $O(10^{-10})$ (or $O(10^{-15})$). In particular, BRes and B_RRes showed that the FSDA attained the prescribed accuracy at the 5th iteration, but the corresponding residual of the low-rank part was still between 10^{-8} and 10^{-9} . So two additional iterations were required to meet the termination condition (40), even if the residual level in B_RRes kept stagnant in the last three iterations. From a structured point of view, it seems that the low-rank part is approaching the critical case while the banded part still lies in the non-critical case. Similarly, $[b_k^g b_k^h b_k^a m_k^h m_k^g]$ indicate that $D_k^G, D_k^H,$ and D_k^A are all block diagonal with block sizes ≤ 6 and the deflation and PTC for the low-rank factors are effective. Moreover, t_k shows that the CPU times at the current iteration were less than twice that of the previous iteration when $k \geq 3$.

Table 6. Residuals, spans of columns, and CPU times at $\zeta = 90$ in Example 3.

k	BRes	B_RRes	LRes	LR_RRes	\tilde{r}_k	$[b_k^g b_k^h b_k^a m_k^h m_k^g]$	t_k
$N = 13,200 \quad \zeta = 90 \quad \tau_g = \tau_h = 10^{-16} \quad m_{\max} = 2000$							
1	1.02×10^3	1.42×10^{-2}	$1.59 \times 10^3 *$	$2.12 \times 10^{-2} *$	$3.35 \times 10^{-2} *$	[6 6 6 29 29]	1.05
2	2.33×10^0	3.25×10^{-5}	$3.04 \times 10^{-1} *$	$4.24 \times 10^{-6} *$	$3.68 \times 10^{-5} *$	[6 6 6 66 66]	5.03
3	6.19×10^{-3}	8.64×10^{-8}	$4.38 \times 10^{-2} *$	$6.11 \times 10^{-7} *$	$6.99 \times 10^{-7} *$	[6 6 6 76 76]	82.23
4	6.09×10^{-7}	8.49×10^{-12}	3.45×10^{-3}	4.81×10^{-8}	4.81×10^{-8}	[6 6 6 100 101]	162.04
5	1.00×10^{-9}	1.40×10^{-14}	9.49×10^{-4}	1.32×10^{-8}	1.32×10^{-8}	[6 6 6 169 170]	248.86
6	1.00×10^{-9}	1.40×10^{-14}	1.01×10^{-5}	1.41×10^{-10}	1.41×10^{-10}	[6 6 6 225 256]	355.07
7	1.00×10^{-9}	1.40×10^{-14}	9.45×10^{-10}	1.31×10^{-14}	2.72×10^{-14}	[6 6 6 225 256]	449.81
$N = 26,400 \quad \zeta = 90 \quad \tau_g = \tau_h = 10^{-16} \quad m_{\max} = 2000$							
1	1.44×10^3	1.42×10^{-2}	$1.61 \times 10^3 *$	$2.21 \times 10^{-2} *$	$3.63 \times 10^{-2} *$	[6 6 6 29 29]	3.89
2	3.29×10^0	3.25×10^{-5}	$3.04 \times 10^{-1} *$	$3.00 \times 10^{-6} *$	$3.55 \times 10^{-5} *$	[6 6 6 66 66]	14.24
3	8.76×10^{-3}	8.64×10^{-8}	$4.38 \times 10^{-2} *$	$4.32 \times 10^{-7} *$	$5.19 \times 10^{-7} *$	[6 6 6 76 76]	554.22
4	8.61×10^{-7}	8.49×10^{-12}	3.45×10^{-3}	3.40×10^{-8}	3.40×10^{-8}	[6 6 6 100 101]	1100.79
5	1.42×10^{-9}	1.40×10^{-14}	9.49×10^{-4}	9.35×10^{-9}	9.35×10^{-9}	[6 6 6 169 170]	1667.77
6	1.42×10^{-9}	1.40×10^{-14}	1.01×10^{-5}	1.00×10^{-10}	1.00×10^{-10}	[6 6 6 210 234]	2286.67
7	1.42×10^{-9}	1.40×10^{-14}	9.46×10^{-10}	9.33×10^{-15}	2.33×10^{-14}	[6 6 6 210 234]	2924.54
$N = 39,600 \quad \zeta = 90 \quad \tau_g = \tau_h = 10^{-16} \quad m_{\max} = 2000$							
1	1.76×10^3	1.42×10^{-2}	$1.61 \times 10^3 *$	$2.21 \times 10^{-2} *$	$3.63 \times 10^{-2} *$	[6 6 6 29 29]	7.49
2	4.03×10^0	3.25×10^{-5}	$3.04 \times 10^{-1} *$	$2.45 \times 10^{-6} *$	$3.49 \times 10^{-5} *$	[6 6 6 66 66]	28.02
3	1.07×10^{-2}	8.64×10^{-8}	$4.38 \times 10^{-2} *$	$3.53 \times 10^{-7} *$	$4.39 \times 10^{-7} *$	[6 6 6 76 76]	564.66
4	1.05×10^{-6}	8.49×10^{-12}	3.45×10^{-3}	2.78×10^{-8}	2.78×10^{-8}	[6 6 6 100 101]	1206.85
5	1.74×10^{-9}	1.40×10^{-14}	9.49×10^{-4}	7.64×10^{-9}	7.64×10^{-9}	[6 6 6 169 170]	1929.52
6	1.74×10^{-9}	1.40×10^{-14}	1.01×10^{-5}	8.19×10^{-11}	8.19×10^{-11}	[6 6 5 209 234]	3553.12
7	1.74×10^{-9}	1.40×10^{-14}	9.48×10^{-10}	7.63×10^{-15}	2.17×10^{-14}	[6 6 0 209 234]	5806.44

We further compare numerical performances between the FSDA and SDA_HODLR for large-scale problems. Different values of ζ have been tried and the compared numerical behaviors of both algorithms are analogous. We list the results of $\zeta = 98$ and $\zeta = 250$ in Table 7, where one can see that the FSDA requires less iterations and CPU time to satisfy the stop criterion than the SDA_HODLR. Particularly, the SDA_HODLR depleted all memory at $N = 39,600$ and did not yield any numerical results (denoted by “—”). The symbols “*” in the SDA_HODLR column represent no related records for bandwidths and column numbers of the low-rank factors.

Table 7. Numerical results between FSDA and SDA_HODLR of Example 3. The symbol * stands for no related records.

N	13,200		26,400		39,600		
	FSDA	SDA_HD	FSDA	SDA_HD	FSDA	SDA_HD	
$\zeta = 98$	b_k^s	[6 6 6 6]	*	[6 6 6 6]	*	[6 6 6 6]	*
	b_k^h	[6 6 6 6]	*	[6 6 6 6]	*	[6 6 6 6]	*
	b_k^a	[6 6 6 6]	*	[6 6 6 6]	*	[6 6 6 6]	*
	m_k^h	[29 66 77 101]	*	[29 66 77 102]	*	[29 66 77 102]	*
	m_k^s	[29 66 77 102]	*	[29 66 77 103]	*	[29 66 77 102]	*
	IT.	4	5	4	5	4	—
	RES.	8.01×10^{-12}	1.64×10^{-12}	7.42×10^{-12}	1.50×10^{-14}	6.22×10^{-12}	—
	CPU	162.18	1130.93	1148.34	18,832.71	1246.78	—
$\zeta = 250$	b_k^s	[6 6 6]	*	[6 6 6]	*	[6 6 6]	*
	b_k^h	[6 6 6]	*	[6 6 6]	*	[6 6 6]	*
	b_k^a	[6 6 6]	*	[6 6 6]	*	[6 6 6]	*
	m_k^h	[29 66 69]	*	[29 66 69]	*	[29 66 69]	*
	m_k^s	[29 66 73]	*	[29 66 71]	*	[29 66 73]	*
	IT.	3	3	3	3	3	—
	RES.	1.75×10^{-12}	1.73×10^{-12}	2.54×10^{-12}	1.74×10^{-12}	3.62×10^{-12}	—
	CPU	80.96	655.69	536.76	15,322.53	634.70	—

7. Conclusions

The stabilizing solution of the discrete-time algebraic Riccati Equation (DARE) from the fractional system, with high-rank non-linear term G and constant term H , is not of numerical low-rank. The structure-preserving doubling algorithm (SDA_h) proposed in [18] is no longer applicable for large scale problems. In some applications, such as in power systems, the state matrix A is of banded-plus-low-rank, and in those cases SDA can be further developed to the factorized structure-preserving doubling algorithm (FSDA) to solve large scale DAREs with high-rank non-linear and constant terms. Under the assumption that G and H are positive semidefinite and D^G and D^H are banded matrices with banded inverse (BMBI), we presented the iterative scheme of FSDA, as well as the convergence of the banded and the low-ranked parts. A deflation process and the technique of PCT are subsequently proposed to efficiently control the growth of the number of columns of low-rank factors. Numerical experiments have demonstrated that the FSDA always reaches the economical pre-terminating condition associated with the banded part before the real terminating condition related to the low-rank part, yielding good approximated solutions $H_k = D_k^H + L_k^H K_k^H (L_k^H)^\top$ and $G_k = D_k^G + L_k^G K_k^G (L_k^G)^\top$ to the DARE and its dual, respectively. Moreover, our FSDA is superior to the existing SDA_HODLR on the CPU time for large-scale DAREs. For future work, the computation of the stabilizing solution for CAREs might be further investigated. This will be more complicated as the Cayley transformation is incorporated and the selection of the corresponding parameter does not seem easy. In addition, other sparse structures of A and high-rank H and G might be investigated.

Author Contributions: Conceptualization, B.Y.; methodology, B.Y.; software, N.D.; validation, N.D.; and formal analysis, B.Y. All authors have read and agreed to the final version of this manuscript.

Funding: This work was supported in part by the NSF of China (11801163), the NSF of Hunan Province (2021JJ50032, 2023JJ50165), the foundation of Education Department of Hunan Province (HNJG-2021-0129) and Degree & Postgraduate Education Reform Project of Hunan University of Technology and Hunan Province (JG2315, 2023JGYB210).

Acknowledgments: Part of the work occurred when the first author visited Monash University. The authors also thank the editor and three anonymous referees for their helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

$$\begin{aligned}
 L_2^G &= \left[\begin{array}{c|c|c|c|c} L_1^G & L_{11}^A & D_1^{AGH}L_1^G & D_1^{AGHG}L_1^H & D_1^{AGHG}L_{21}^A \end{array} \right] \\
 &= \left[\begin{array}{c|c|c|c|c} L_{10}^A, D_0^{AGHG}L_{20}^A & L_{10}^A, D_0^{AGH}L_{10}^A & D_1^{AGH}L_{10}^A, D_1^{AGH}D_0^{AGHG}L_{20}^A & D_1^{AGHG}L_{20}^A, D_1^{AGHG}D_0^{AGH}L_{10}^A & D_1^{AGHG}L_{20}^A, D_1^{AGHG}D_0^{AGH}L_{10}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c|c} D_0^{AGHG}L_{20}^A & L_{10}^A, D_0^{AGH}L_{10}^A & D_1^{AGH}L_{10}^A, D_1^{AGH}D_0^{AGHG}L_{20}^A & D_1^{AGHG}L_{20}^A, D_1^{AGHG}D_0^{AGH}L_{10}^A & D_1^{AGHG}D_0^{AGH}L_{10}^A \end{array} \right] := L_2^{Gd},
 \end{aligned}$$

$$\begin{aligned}
 L_{12}^A &= \left[\begin{array}{c|c|c|c} L_{11}^A & D_1^{AGH}L_1^G & D_1^{AGHG}L_1^H & D_1^{AGHG}L_{21}^A \end{array} \right] \\
 &= \left[\begin{array}{c|c|c|c} L_{10}^A, D_0^{AGH}L_{10}^A & D_1^{AGH}L_{10}^A, D_1^{AGH}D_0^{AGHG}L_{20}^A & D_1^{AGHG}L_{20}^A, D_1^{AGHG}D_0^{AGH}L_{10}^A & D_1^{AGH}L_{10}^A, D_1^{AGH}D_0^{AGHG}L_{20}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c} L_{10}^A, D_0^{AGH}L_{10}^A & D_1^{AGH}L_{10}^A, D_1^{AGH}D_0^{AGHG}L_{20}^A & D_1^{AGHG}L_{20}^A, D_1^{AGHG}D_0^{AGH}L_{10}^A & D_1^{AGH}D_0^{AGHG}L_{20}^A \end{array} \right] := L_{12}^{Ad},
 \end{aligned}$$

$$\begin{aligned}
 L_2^H &= \left[\begin{array}{c|c|c|c|c} L_1^H & L_{21}^A & D_1^{A^\top HG}L_1^H & D_1^{A^\top HGH}L_1^G & D_1^{A^\top HGH}L_{11}^A \end{array} \right] \\
 &= \left[\begin{array}{c|c|c|c|c} L_{20}^A, D_0^{A^\top HGH}L_{10}^A & L_{20}^A, D_0^{A^\top HG}L_{20}^A & D_1^{A^\top HG}L_{20}^A, D_1^{A^\top HG}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HGH}L_{20}^A, D_1^{A^\top HGH}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HGH}L_{20}^A, D_1^{A^\top HGH}D_0^{A^\top HGH}L_{10}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c|c} D_0^{A^\top HGH}L_{10}^A & L_{20}^A, D_0^{A^\top HG}L_{20}^A & D_1^{A^\top HG}L_{20}^A, D_1^{A^\top HG}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HGH}L_{20}^A, D_1^{A^\top HGH}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HGH}D_0^{A^\top HGH}L_{10}^A \end{array} \right] := L_2^{Hd},
 \end{aligned}$$

$$\begin{aligned}
 L_{22}^A &= \left[\begin{array}{c|c|c|c} L_{21}^A & D_1^{A^\top HG}L_1^H & D_1^{A^\top HGH}L_1^G & D_1^{A^\top HG}L_{21}^A \end{array} \right] \\
 &= \left[\begin{array}{c|c|c|c} L_{20}^A, D_0^{A^\top HG}L_{20}^A & D_1^{A^\top HG}L_{20}^A, D_1^{A^\top HG}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HGH}L_{20}^A, D_1^{A^\top HGH}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HG}L_{20}^A, D_1^{A^\top HG}D_0^{A^\top HG}L_{20}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c} L_{20}^A, D_0^{A^\top HG}L_{20}^A & D_1^{A^\top HG}L_{20}^A, D_1^{A^\top HG}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HGH}L_{20}^A, D_1^{A^\top HGH}D_0^{A^\top HGH}L_{10}^A & D_1^{A^\top HG}D_0^{A^\top HG}L_{20}^A \end{array} \right] := L_{22}^{Ad}.
 \end{aligned}$$

Matrices L_1^G , L_{11}^A , L_1^H and L_{21}^A are actually the deflated, truncated and compressed low-rank factors L_1^{Gdt} , L_{11}^{Adt} , L_1^{Hdt} and L_{21}^{Adt} , respectively. We omit the superscript “dt” for the simpler notation.

Appendix B

$$\begin{aligned}
 L_k^G &= \left[\begin{array}{c|c|c|c|c} L_{k-1}^G & L_{1,k-1}^A & D_{k-1}^{AGH}L_{k-1}^G & D_{k-1}^{AGHG}L_{k-1}^H & D_{k-1}^{AGHG}L_{2,k-1}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c|c} D_0^{AGHG}L_{20}^A, D_1^{AGHG}D_0^{A^\top HG}L_{20}^A, \dots, D_{k-2}^{AGHG}\Pi_{i=k-3}^0 D_i^{A^\top HG}L_{20}^A & L_{1,k-1}^A & D_{k-1}^{AGH}L_{k-1}^G & D_{k-1}^{AGHG}L_{k-1}^H & D_{k-1}^{AGHG}\Pi_{i=k-2}^0 D_i^{A^\top HG}L_{20}^A \end{array} \right] := L_k^{Gd}, \tag{A1}
 \end{aligned}$$

$$\begin{aligned}
 L_{1,k}^A &= \left[\begin{array}{c|c|c|c} L_{1,k-1}^A & D_{k-1}^{AGH}L_{k-1}^G & D_{k-1}^{AGHG}L_{k-1}^H & D_{k-1}^{AGH}L_{1,k-1}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c} L_{1,k-1}^A & D_{k-1}^{AGH}L_{k-1}^G & D_{k-1}^{AGHG}L_{k-1}^H & \Pi_{i=k-1}^0 D_i^{AGH}L_{10}^A \end{array} \right] := L_{1,k}^{Ad}, \tag{A2}
 \end{aligned}$$

$$\begin{aligned}
 L_k^H &= \left[\begin{array}{c|c|c|c|c} L_{k-1}^H & L_{2,k-1}^A & D_{k-1}^{A^\top HG}L_{k-1}^H & D_{k-1}^{A^\top HGH}L_{k-1}^G & D_{k-1}^{A^\top HGH}L_{1,k-1}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c|c} D_0^{A^\top HGH}L_{10}^A, D_1^{A^\top HGH}D_0^{AGH}L_{10}^A, \dots, D_{k-2}^{A^\top HGH}\Pi_{i=k-3}^0 D_i^{AGH}L_{10}^A & L_{2,k-1}^A & D_{k-1}^{A^\top HG}L_{k-1}^H & D_{k-1}^{A^\top HGH}L_{k-1}^G & D_{k-1}^{A^\top HGH}\Pi_{i=k-2}^0 D_i^{AGH}L_{10}^A \end{array} \right] := L_k^{Hd}, \tag{A3}
 \end{aligned}$$

$$\begin{aligned}
 L_{2,k}^A &= \left[\begin{array}{c|c|c|c} L_{2,k-1}^A & D_{k-1}^{A^\top HG}L_{k-1}^H & D_{k-1}^{A^\top HGH}L_{k-1}^G & D_{k-1}^{A^\top HG}L_{2,k-1}^A \end{array} \right] \\
 \xrightarrow{d} & \left[\begin{array}{c|c|c|c} L_{2,k-1}^A & D_{k-1}^{A^\top HG}L_{k-1}^H & D_{k-1}^{A^\top HGH}L_{k-1}^G & \Pi_{i=k-1}^0 D_i^{A^\top HG}L_{20}^A \end{array} \right] := L_{2,k}^{Ad}. \tag{A4}
 \end{aligned}$$

Matrices $L_{k-1}^G, L_{1,k-1}^A, L_{k-1}^H$ and $L_{2,k-1}^A$ are actually the deflated, truncated and compressed low-rank factors $L_{k-1}^{Gdt}, L_{1,k-1}^{Adt}, L_{k-1}^{Hdt}$ and $L_{2,k-1}^{Adt}$, respectively. We omit the superscript “dt” for convenience.

Appendix C. Description for the Deflation of L_3^G

$$L_3^G = \begin{bmatrix} m_2^g & m_2^{a1} & m_2^g & m_2^h & m_2^{a2} \\ L_2^G & L_{12}^A & D_2^{AGH} L_2^G & D_2^{AGHG} L_2^H & D_2^{AGHG} L_{22}^A \end{bmatrix} N$$

$$= \left[\begin{array}{c} D_0^{AGHG} L_{20}^A, L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGHG} D_0^T H G L_{20}^A, \\ L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGH} D_0^{AGH} L_{10}^A, \\ D_2^{AGH} L_2^A, \\ D_2^{AGHG} (D_0^T H G L_{10}^A, L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G H D_0^{AGH} L_{10}^A) \\ D_2^{AGHG} (L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G D_0^T H G L_{20}^A) \\ \mathbf{D_0^{AGHG} L_{20}^A}, \\ L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGH} D_0^{AGH} L_{10}^A, \\ D_2^{AGH} L_2^A, \\ D_2^{AGHG} (D_0^T H G L_{10}^A, L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G H D_0^{AGH} L_{10}^A) \\ D_2^{AGHG} D_1^T H G D_0^T H G L_{20}^A \end{array} \right] \tag{A5}$$

$$\left[\begin{array}{c} L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGH} D_0^{AGH} L_{10}^A, \\ D_2^{AGH} L_2^A, \\ D_2^{AGHG} (L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G D_0^T H G L_{20}^A) \end{array} \right] \tag{A6}$$

$$\left[\begin{array}{c} D_2^{AGHG} (D_0^T H G L_{10}^A, L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G H D_0^{AGH} L_{10}^A) \\ D_2^{AGHG} (L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G D_0^T H G L_{20}^A) \end{array} \right] \tag{A7}$$

$$\xrightarrow{d} \left[\begin{array}{c} D_0^{AGHG} L_{20}^A, \\ L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGH} D_0^{AGH} L_{10}^A, \\ D_2^{AGH} L_2^A, \\ D_2^{AGHG} (D_0^T H G L_{10}^A, L_{20}^A, D_0^T H G L_{20}^A, D_1^T H G L_{20}^A, D_1^T H G D_0^T H G H L_{10}^A, D_1^T H G H L_{10}^A, D_1^T H G H D_0^{AGHG} L_{20}^A, D_1^T H G H D_0^{AGH} L_{10}^A) \\ D_2^{AGHG} D_1^T H G D_0^T H G L_{20}^A \end{array} \right] \tag{A8}$$

$$= \left[D_0^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{20}^A \mid L_{12}^A \mid D_2^{AGH} L_2^G \mid D_2^{AGHG} L_2^H \mid D_2^{AGHG} D_1^T H G D_0^T H G L_{20}^A \right] N := L_3^{Gd}.$$

After the previous deflation, there are $m_2^g - 2m^a$ columns in L_2^G and L_{12}^A (items marked with bold type in (A2) and (A3)) and $m_2^a - m^a$ columns (items marked with bold type in (A4) and (A5)) in $D_2^{AGHG} L_2^H$ and $D_2^{AGHG} L_{22}^A$ are identical. Then, one can remove columns of $L_2^G(:, m^a + 1 : m_2^g - m^a)$ in (A2) and $D_2^{AGHG} L_{22}^A(:, 1 : m_2^a - m^a)$ in (A5) (i.e., items with bold type in (A2) and (A5)), keep columns of $L_{12}^A(:, 1 : m_2^g - 2m^a)$ in (A3) and $D_2^{AGHG} L_2^H(:, m_2^h - m_2^{a2} + 1 : m_2^h - m^a)$ in (A4) (i.e., items with bold type in (A3) and (A4)), respectively. Then there are two matrices with each of order $N \times m^a$ are left in L_2^G and only one matrix of order $N \times m^a$ left in $D_2^{AGHG} L_{22}^A$.

Note that matrices L_2^G, L_{12}^A, L_2^H and L_{22}^A are actually the deflated, truncated and compressed low-rank factors $L_2^{Gdt}, L_{12}^{Adt}, L_2^{Hdt}$ and L_{22}^{Adt} , respectively.

Appendix D. Description for the Deflation of L_{13}^A

$$L_{13}^A = \begin{bmatrix} m_2^{a1} & m_2^g & m_2^h & m_2^{a1} \\ L_{12}^A & D_2^{AGH} L_2^G & D_2^{AGHG} L_2^H & D_2^{AGH} L_{12}^A \end{bmatrix} N$$

$$= \left[\begin{array}{c} L_{12}^A, \\ D_2^{AGH} (D_0^{AGHG} L_{20}^A, L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGHG} D_0^T H G L_{20}^A), \\ D_2^{AGHG} L_2^H, \\ D_2^{AGH} (L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGH} D_0^{AGH} L_{10}^A) \end{array} \right] \tag{A9}$$

$$\left[\begin{array}{c} L_{12}^A, \\ D_2^{AGH} (D_0^{AGHG} L_{20}^A, L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGHG} D_0^T H G L_{20}^A), \\ D_2^{AGHG} L_2^H, \\ D_2^{AGH} D_1^T H G D_0^T H G L_{20}^A \end{array} \right] \tag{A10}$$

$$\xrightarrow{d} \left[\begin{array}{c} L_{12}^A, \\ D_2^{AGH} (D_0^{AGHG} L_{20}^A, L_{10}^A, D_0^{AGH} L_{10}^A, D_1^{AGH} L_{10}^A, D_1^{AGH} D_0^{AGHG} L_{20}^A, D_1^{AGHG} L_{20}^A, D_1^{AGHG} D_0^T H G L_{10}^A, D_1^{AGHG} D_0^T H G L_{20}^A), \\ D_2^{AGHG} L_2^H, \\ D_2^{AGH} D_1^T H G D_0^T H G L_{20}^A \end{array} \right] \tag{A11}$$

$$= \left[L_{12}^A \mid D_2^{AGH} L_2^G \mid D_2^{AGHG} L_2^H \mid D_2^{AGH} D_1^T H G D_0^T H G L_{20}^A \right] N := L_{13}^{Ad}.$$

To deflate L_{13}^A , columns of $D_2^{AGH}L_{12}^A(:, 1 : m_2^{a_1} - m^a)$ are removed (i.e., items marked with bold type in (A7)) but columns of $D_2^{AGH}L_2^G(:, m_2^g - m_2^{a_1} + 1 : m_2^g - m^a)$ (i.e., items marked with bold type in (A6)) are retained in L_{12}^A . So only one matrix of order $N \times m^a$ is left in $D_2^{AGH}L_{12}^A$, i.e., the last item $\Pi_{i=2}^0 D_i^{AGH}L_{10}^A$ in (A8).

Note that matrices L_2^G , L_{12}^A and L_2^H are actually the deflated, truncated, and compressed low-rank factors L_2^{Gdt} , L_{12}^{Adt} , and L_2^{Hdt} , respectively.

Appendix E

Table A1. Complexity and memory requirement at k -th iteration in the FSDA.

Items	Flops	Memory
Banded part		
$D_k^{AGH}, D_k^{A^T HG} *$	$4N(2b_{k-1}^{hg} + 1)^2 + b_{k-1}^{hg} b_{k-1}^a$	$2N(2b_{k-1}^{hga} + 1)$
D_k^G, D_k^H, D_k^A	$4N(2b_{k-1}^g + 1)(2b_{k-1}^{hga} + 1)$	$2N(2b_{k-1}^{hga} + 1)$
Low-rank part and kernels		
$D_{k-1}^{AGH}L_{k-1}^G, D_{k-1}^{AGHG}L_{k-1}^H, D_{k-1}^{AGHG}L_{2,k-1}^A$	$2Nb_{k-1}^{hga}(m_{k-1}^g + m_{k-1}^h + m_{k-1}^a)$	$(m_{k-1}^g + m_{k-1}^h + m_{k-1}^a)N$
$D_{k-1}^{A^T HG}L_{k-1}^H, D_{k-1}^{A^T HG}L_{k-1}^G, D_{k-1}^{A^T HG}L_{1,k-1}^A$	$2Nb_{k-1}^{hga}(m_{k-1}^g + m_{k-1}^h + m_{k-1}^a)$	$(m_{k-1}^g + m_{k-1}^h + m_{k-1}^a)N$
$\Theta_{k-1}^H, \Theta_{k-1}^G, \Theta_{k-1}^{HG}$	$2N(b_{k-1}^{hg}(m_{k-1}^h + m_{k-1}^g) + b_{k-1}^{hg}m_{k-1}^g + (m_{k-1}^h)^2 + (m_{k-1}^g)^2 + m_{k-1}^h m_{k-1}^g)$	$(m_{k-1}^h)^2 + (m_{k-1}^g)^2 + m_{k-1}^h m_{k-1}^g$
$\Theta_{k-1}^A, \Theta_{1,k-1}^A, \Theta_{2,k-1}^A$	$2N(2b_{k-1}^{hg}m_{k-1}^a + b_{k-1}^{hg}m_{k-1}^a + 3(m_{k-1}^a)^2)$	$3(m_{k-1}^a)^2$
$\Theta_{1,k-1}^{AH}, \Theta_{1,k-1}^{AG}$	$2N(b_{k-1}^{hg}(m_{k-1}^h + m_{k-1}^g) + m_{k-1}^a(m_{k-1}^h + m_{k-1}^g))$	$m_{k-1}^a(m_{k-1}^h + m_{k-1}^g)$
$\Theta_{2,k-1}^{AH}, \Theta_{2,k-1}^{AG}$	$2N(b_{k-1}^{hg}(m_{k-1}^h + m_{k-1}^g) + m_{k-1}^a(m_{k-1}^h + m_{k-1}^g))$	$m_{k-1}^a(m_{k-1}^h + m_{k-1}^g)$
K_{k-1}^{AGHG}	$(m_{k-1}^a)^2(m_{k-1}^h + m_{k-1}^g) + m_{k-1}^a(m_{k-1}^h + m_{k-1}^g)^2$	$m_{k-1}^a(m_{k-1}^h + m_{k-1}^g)$
$K_{k-1}^{AGHGA^T}, K_{k-1}^{A^T HGHA}, K_{k-1}^{AGHA}$	$6(m_{k-1}^a)^2(2m_{k-1}^a + m_{k-1}^h + m_{k-1}^g)$	$3(m_{k-1}^a)^2$
$K_{k-1}^{A^T HGH}$	$2(m_{k-1}^a)(m_{k-1}^a + m_{k-1}^h)(m_{k-1}^a + m_{k-1}^h + m_{k-1}^g)$	$m_{k-1}^a(m_{k-1}^h + m_{k-1}^g)$
$K_{k-1}^{AGH}, K_{k-1}^{A^T GH}$	$2(m_{k-1}^a)(m_{k-1}^a + m_{k-1}^h)^2$	$2m_{k-1}^a(m_{k-1}^h + m_{k-1}^g)$
$K_{k-1}^{GH*}, K_{k-1}^{GHG^T}, K_{k-1}^{HGH}$	$8(m_{k-1}^a + m_{k-1}^g)^3/3$	$3(m_{k-1}^h + m_{k-1}^g)^2$
Q_k^G, Q_k^{H**}	$4(m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)^2(N - m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)$	$(r_k^h + r_k^g)N$
$U_k^G, U_k^H,$	$4(m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)r_{k-1}^g(N - m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)$	$(r_k^g + r_k^h) \times m_{k-1}^{hga}$
K_k^{Gdt}	$12(m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)^2 r_{k-1}^g$	$(m_k^g)^2$
K_k^{Hdt}	$12(m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)^2 r_{k-1}^h$	$(m_k^h)^2$
K_k^{Adt}	$6(m_{k-1}^a + m_{k-1}^g + m_{k-1}^h)^2 (r_{k-1}^g + r_{k-1}^h)$	$m_k^g m_k^h$
Residual part		
$(D_0^A)^T \tilde{D}_k^{HGH}L_{10}^A, (D_0^A)^T \tilde{D}_k^{HG}L_k^H$	$2b_k^{hg}(m^a + m_k^h)N$	$(m_k^h + m^a)N$
$(L_k^H)^T \tilde{D}_k^{GHG}L_k^H$	$2b_k^{hg}(m^a + m_k^h)N$	$(m_k^h)^2$
$\tilde{K}_k^H *$	$8(m_k^h)^2/3$	$(m_k^h)^2$
$\tilde{K}_k^{A^T HG}$	$2b_k^{hg}(m^a + m_k^h)N$	$m^a m_k^h$
$\tilde{K}_k^{A^T HGHA}$	$2m^a(b_k^{hg} + m^a)N + 2(m^a)^2 m_k^h$	$(m^a)^2$
$Q_k^R **$	$2(m^a + 2m_k^h)^2(N - m^a - 2m_k^h)$	$r_k^r N$
U_k^R	$2(m^a + 2m_k^h)r_k^r(N - m^a - 2m_k^h)$	$r_k^r(m^a + 2m_k^h)$
$U_k^R K_k^{Rd} (U_k^R)^T$	$2(m^a + 2m_k^h)r_k^r(r_k^r + m^a + 2m_k^h)$	$(r_k^r)^2$

* LU factorization and Gaussian elimination is used [42]. ** Householder QR decomposition is used [12].

References

- Nosrati, K.; Shafiee, M. On the convergence and stability of fractional singular Kalman filter and Riccati equation. *J. Frankl. Inst.* **2020**, *357*, 7188–7210.
- Trujillo, J.J.; Ungureanu, V.M. Optimal control of discrete-time linear fractional-order systems with multiplicative noise. *Int. J. Control* **2018**, *91*, 57–69. [CrossRef]
- Podlubny, I. *Fractional Differential Equations*; Academic Press: New York, NY, USA, 1999.
- Benner, P.; Fassbender, H. The symplectic eigenvalue problem, the butterfly form, the SR algorithm, and the Lanczos method. *Linear Algebra Appl.* **1998**, *275–276*, 19–47.

5. Chen, C.-R. A structure-preserving doubling algorithm for solving a class of quadratic matrix equation with M -matrix. *Electron. Res. Arch.* **2022**, *30*, 574–581.
6. Chu, E.K.-W.; Fan, H.-Y.; Lin, W.-W. A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations. *Linear Algebra Appl.* **2005**, *396*, 55–80.
7. Chu, E.K.-W.; Fan, H.-Y.; Lin, W.-W.; Wang, C.-S. A structure-preserving doubling algorithm for periodic discrete-time algebraic Riccati equations. *Int. J. Control* **2004**, *77*, 767–788.
8. Kleinman, D. On an iterative technique for Riccati equation computations. *IEEE Trans. Autom. Control* **1968**, *13*, 114–115. [[CrossRef](#)]
9. Lancaster, P.; Rodman, L. *Algebraic Riccati Equations*; Clarendon Press: Oxford, UK, 1995.
10. Laub, A.J. A Schur method for solving algebraic Riccati equation. *IEEE Trans. Autom. Control* **1979**, *AC-24*, 913–921.
11. Li, T.-X.; Chu, D.-L. A structure-preserving algorithm for semi-stabilizing solutions of generalized algebraic Riccati equations. *Electron. Trans. Numer. Anal.* **2014**, *41*, 396–419.
12. Mehrmann, V.L. *The Autonomous Linear Quadratic Control Problem*; Lecture Notes in Control and Information Sciences; Springer: Berlin, Germany, 1991; Volume 163.
13. Mohammad, I. Fractional polynomial approximations to the solution of fractional Riccati equation. *Punjab Univ. J. Math.* **2019**, *51*, 123–141.
14. Tyvordiy, D.A. Hereditary Riccati equation with fractional derivative of variable order. *J. Math. Sci.* **2021**, *253*, 564–572. [[CrossRef](#)]
15. Yu, B.; Li, D.-H.; 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. [[CrossRef](#)]
16. Benner, P.; Saak, J.A. Galerkin-Newton-ADI method for solving large-scale algebraic Riccati equations. In *DFG Priority Programme 1253 “Optimization with Partial Differential Equations”*; Preprint SPP1253-090; DFG: Bonn, Germany, 2010.
17. Chu, E.K.-W.; Weng, P.C.-Y. Large-scale discrete-time algebraic Riccati equations—Doubling algorithm and error analysis. *J. Comput. Appl. Math.* **2015**, *277*, 115–126. [[CrossRef](#)]
18. Yu, B.; Fan, H.-Y.; Chu, E.K.-W. Large-scale algebraic Riccati equations with high-rank constant terms. *J. Comput. Appl. Math.* **2019**, *361*, 130–143. [[CrossRef](#)]
19. Martins, N.; Lima, L.; Pinto, H. Computing dominant poles of power system transfer functions. *IEEE Trans. Power Syst.* **1996**, *11*, 162–170. [[CrossRef](#)]
20. Freitas, F.D.; Martins, N.; Varricchio, S.L.; Rommes, J.; Veliz, F.C. Reduced-Order Transfer Matrices from RLC Network Descriptor Models of Electric Power Grids. *IEEE Trans. Power Syst.* **2011**, *26*, 1905–1916. [[CrossRef](#)]
21. Rommes, J.; Martins, N. Efficient computation of multivariable transfer function dominant poles using subspace acceleration. *IEEE Trans. Power Syst.* **2006**, *21*, 1471–1483. [[CrossRef](#)]
22. Dahmen, W.; Micchelli, C.C. Banded matrices with banded inverses, II: Locally finite decomposition of spline spaces. *Constr. Approx.* **1993**, *9*, 263–281. [[CrossRef](#)]
23. Cantero, M.J.; Moral, L.; Velázquez, L. Five-diagonal matrices and zeros of orthogonal polynomials on the unit circle. *Linear Algebra Appl.* **2003**, *362*, 29–56. [[CrossRef](#)]
24. Kimura, H. Generalized Schwarz form and lattice-ladder realizations of digital filters. *IEEE Trans. Circuits Syst.* **1985**, *32*, 1130–1139. [[CrossRef](#)]
25. Kavcic, A.; Moura, J. Matrices with banded inverses: Inversion algorithms and factorization of Gauss–Markov processes. *IEEE Trans. Inf. Theory* **2000**, *46*, 1495–1509. [[CrossRef](#)]
26. Strang, G. Fast transforms: Banded matrices with banded inverses. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 12413–12416. [[CrossRef](#)] [[PubMed](#)]
27. Strang, G. Groups of banded matrices with banded inverses. *Proc. Am. Math. Soc.* **2011**, *139*, 4255–4264. [[CrossRef](#)]
28. Strang, G.; Nguyen, T. *Wavelets and Filter Banks*; Wellesley–Cambridge Press: Cambridge, UK, 1996.
29. Olshevsky, V.; Zhlobich, P.; Strang, G. Green’s matrices. *Linear Algebra Appl.* **2010**, *432*, 218–241. [[CrossRef](#)]
30. Grasedyck, L.; Hackbusch, W.; Khoromskij, B.N. Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing* **2003**, *70*, 121–165. [[CrossRef](#)]
31. Kressner, D.; Krschner, P.; Massei, S. Low-rank updates and divide-and-conquer methods for quadratic matrix equations. *Numer. Algorithms* **2020**, *84*, 717–741. [[CrossRef](#)]
32. Benner, P.; Saak, J. A Semi-Discretized Heat Transfer Model for Optimal Cooling of Steel Profiles. In *Dimension Reduction of Large-Scale Systems*; Benner, P., Sorensen, D.C., Mehrmann, V., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2005; Volume 45.
33. Korvink, G.; Rudnyi, B. Oberwolfach Benchmark Collection. In *Dimension Reduction of Large-Scale Systems*; Benner, P., Sorensen, D.C., Mehrmann, V., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2005; Volume 45.
34. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; Johns Hopkins University Press: Baltimore, MD, USA, 1996.
35. Huang, C.-M.; Li, R.-C.; Lin, W.-W. *Structure-Preserving Doubling Algorithms for Nonlinear Matrix Equations*; SIAM: Washington, DC, USA, 2018.
36. Lin, W.-W.; Xu, S.-F. Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations. *SIAM J. Matrix Anal. Appl.* **2006**, *28*, 26–39. [[CrossRef](#)]

37. Demko, S. Inverses of band matrices and local convergence of spline projections. *SIAM J. Numer. Anal.* **1977**, *14*, 616–619. [[CrossRef](#)]
38. Mathworks. *MATLAB User's Guide*; Mathworks: Natick, MA, USA, 2010.
39. Massei, S.; Palitta, D.; Robol, L. Solving rank structured Sylvester and Lyapunov equations. *SIAM J. Matrix Anal. Appl.* **2018**, *39*, 1564–1590. [[CrossRef](#)]
40. Massei, S.; Robol, L.; Kressner, D. hm-toolbox: Matlab software for HODLR and HSS matrices. *SIAM J. Sci. Comput.* **2020**, *42*, C43–C68. [[CrossRef](#)]
41. Watson, N.; Arrillaga, J. *Power Systems Electromagnetic Transients Simulation*; IET, Digital Library: London, UK, 2003.
42. Arbenz, P.; Gander, W. *A Survey of Direct Parallel Algorithms for Banded Linear Systems*; Tech. Report 221; Departement Informatik, Institut für Wissenschaftliches Rechnen, ETH Zürich: Zurich, Switzerland, 1994.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.