ARTICLE IN PRESS
Computers and Mathematics with Applications xxx (xxxx) xxx

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

An extended block Golub–Kahan algorithm for large algebraic and differential matrix Riccati equations

Z. Asgari [a], F. Toutounian [b,*], E. Babolian [a], E. Tohidi [c]

[a] Department of Mathematics, Faculty of Mathematical Science and Computer, Kharazmi University, Tehran, Iran
[b] Department of Applied Mathematics, Faculty of Mathematical Science, Ferdowsi University of Mashhad, Iran
[c] Department of Mathematics, Kosar University of Bojnord, Iran

A R T I C L E I N F O

Article history:
Received 29 December 2018
Received in revised form 30 October 2019
Accepted 16 November 2019
Available online xxxx

Keywords:
Matrix equations
Extended block Krylov subspace
Golub–Kahan bidiagonalization
Large-scale equations

A B S T R A C T

In this paper we propose a new projection method to solve both large-scale continuous-time matrix Riccati equations and differential matrix Riccati equations. The new approach projects the problem onto an extended block Krylov subspace and gets a low-dimensional equation. We use the block Golub–Kahan procedure to construct the orthonormal bases for the extended Krylov subspaces. For matrix Riccati equations, the reduced problem is then solved by means of a direct Riccati scheme such as the Schur method. When we solve differential matrix Riccati equations, the reduced problem is solved by the Backward Differentiation Formula (BDF) method and the obtained solution is used to build the low rank approximate solution of the original problem. Finally, we give some theoretical results and present numerical experiments.

1. Introduction

In this paper, we consider the large continuous-time algebraic Riccati equation (CARE) of the form

$$A^TX + XA - XBB^TX + C^TC = 0, \tag{1}$$

and the continuous-time differential matrix Riccati equation (DRE) on the time in the interval $[0, T_f]$ of the form

$$\begin{cases} \dot{X}(t) = A^TX(t) + X(t)A - X(t)BB^TX(t) + C^TC, \\ X(0) = X_0, \end{cases} \tag{2}$$

where $A \in \mathbb{R}^{n\times n}$ is assumed to be large, sparse, and nonsingular, $B \in \mathbb{R}^{n\times s}$ and $C \in \mathbb{R}^{p\times n}$ are assumed to be full rank matrices with $p, s \ll n$, X_0 is some given $n \times n$ low rank matrix, X is unknown matrix for Eq. (1) and unknown matrix function for Eq. (2).

Continuous-time algebraic Riccati equation (1) and differential matrix Riccati equation (2) play the fundamental roles in many areas such as control, model reduction problems and many others; see, e.g., [1–10] and references therein. In the last decades, some numerical methods have been proposed for approximating solution of large scale algebraic Riccati equations [11–15]. For continuous-time differential matrix Riccati equation, only a few attempts have been made for large problems, see [16,17].

* Corresponding author.
E-mail addresses: za.asgari93@gmail.com (Z. Asgari), toutouni@math.um.ac.ir (F. Toutounian), babolian@khu.ac.ir (E. Babolian), emrantohidi@gmail.com (E. Tohidi).

https://doi.org/10.1016/j.camwa.2019.11.010

In this paper we present a new projection method that projects the initial problem onto an extended block Krylov subspace. The new projection method builds the orthonormal bases of enriched block Krylov subspaces and allows us to compute low rank approximations to the stabilizing solution of (1) and to obtain a low-dimensional differential matrix Riccati equation. The extended block Krylov subspaces are generated by means of the new extended block Golub and Kahan procedure. In addition, we provide new theoretical analysis of the method and the norm of the residual.

We mention that the Golub and Kahan process first introduced in [18]. In [19], the authors defined the block bidiagonalization based on Golub and Kahan procedure. In the present paper, for given matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n\times p}$, we use the block bidiagonalization procedure of Golub and Kahan and generate the orthonormal bases for the following extended Krylov subspaces:

$$\mathcal{K}^e_{k+1,m}(AA^T, C) = \text{span}\{(AA^T)^{-k}C, \dots, (AA^T)^{-1}C, C, (AA^T)C, \dots, (AA^T)^{m-1}C\},$$
$$\mathcal{K}^e_{k,m+1}(A^TA, A^TC) = \text{span}\{(A^TA)^{-k+1}A^TC, \dots, (A^TA)^{-1}A^TC, A^TC, (A^TA)A^TC, \dots, (A^TA)^m A^TC\},$$

where $k, m \geq 1$, dim $\mathcal{K}^e_{k+1,m}(AA^T, C) \leq (k+m)p$, and dim $\mathcal{K}^e_{k,m+1}(A^TA, A^TC) \leq (k+m)p$.

Evidently, $\mathcal{K}^e_{k+1,m}(AA^T, C)$ and $\mathcal{K}^e_{k,m+1}(A^TA, A^TC)$ can be treated as the usual Krylov subspaces, but with other starting matrices $(AA^T)^{-k}C$ and $(A^TA)^{-k+1}A^TC$, respectively. We do not wish to calculate the matrices $(AA^T)^{-k}C$ and $(A^TA)^{-k+1}A^TC$ (they would be numerically unstable), but instead we are interested in developing a special procedure to obtain the orthonormal bases of the extended Krylov subspaces $\mathcal{K}^e_{k+1,m}(AA^T, C)$ and $\mathcal{K}^e_{k,m+1}(A^TA, A^TC)$. The use of the extended subspaces is justified by the fact that they contain more information than the classical Krylov subspaces since they are enriched by $(AA^T)^{-1}$ and $(A^TA)^{-1}$.

Throughout this paper, we use the following notations. For two $n \times s$ matrices X and Y, we define the following inner product: $\langle X, Y\rangle_F = \text{tr}(X^TY)$, where $\text{tr}(Z)$ denotes the trace of the square matrix Z. The associated norm is the Frobenius norm denoted by $\|.\|_F$. We will use the notation $\langle ., .\rangle_2$ for the usual inner product in $\mathbb{R}^n$ and the associated norm denoted by $\|.\|_2$. Finally, $0_{s\times l}$ will denote the zero matrix in $\mathbb{R}^{s\times l}$, 0_s and I_s will denote the zero and the identity matrix in $\mathbb{R}^{s\times s}$, respectively.

The outline of this paper is as follows. In Section 2, we give a quick overview of the block Golub–Kahan procedure and its properties. In Section 3, we present the extended version of block Golub–Kahan procedure and its properties. In Section 4, we show how to apply the extended block Golub–Kahan procedure to obtain low rank approximate solutions to the continuous-time algebraic equation (1). We give some theoretical results for the residual at each step which does not require the computation of products of large matrices. In Section 5, we use the extended block Golub–Kahan procedure for the numerical resolution of the differential Riccati equation (2). The initial differential Riccati equation is projected onto a block extended Krylov subspace to get a low dimensional differential Riccati equation that is solved by the backward differentiation formula (BDF) method. Section 6 is devoted to some numerical experiments. Finally, we make some concluding remarks in Section 7.

2. The block Golub–Kahan procedure

In this section, we present a brief of the block Bidiag 1 algorithm [19]. This algorithm is the basis for the extended block Golub–Kahan procedure.

The block Bidiag 1 procedure constructs the sets of the $n \times p$ block vectors $V_1, V_2, \dots, V_k$ and $U_1, U_2, \dots, U_k$ such that $V_i^TV_j = 0_p$, $U_i^TU_j = 0_p$, for $i \neq j$, and $V_i^TV_i = I_p$, $U_i^TU_i = I_p$ and after k steps they form the orthonormal bases of $\mathbb{R}^{n\times kp}$.

Block Bidiag 1 (*Starting matrix B; reduction to block lower bidiagonal form*)

$$U_1B_1 = B, \quad V_1A_1 = A^TU_1, \tag{3}$$

$$\left.\begin{aligned} U_{i+1}B_{i+1} &= AV_i - U_iA_i^T, \\ V_{i+1}A_{i+1} &= A^TU_{i+1} - V_iB_{i+1}^T, \end{aligned}\right\} \quad i = 1, 2, \dots, k, \tag{4}$$

where $U_i, V_i \in \mathbb{R}^{n\times p}, B_i, A_i \in \mathbb{R}^{p\times p}$ and $U_1B_1, V_1A_1, U_{i+1}B_{i+1}, V_{i+1}A_{i+1}$ are the QR decompositions of the matrices $B, A^TU_1, AV_i - U_iA_i^T, A^TU_{i+1} - V_iB_{i+1}^T$, respectively. With the definitions

$$\overline{U}_k \equiv [U_1, U_2, \dots, U_k], \qquad \overline{V}_k \equiv [V_1, V_2, \dots, V_k], \qquad T_k \equiv \begin{bmatrix} A_1^T & & & \\ B_2 & A_2^T & & \\ & \ddots & \ddots & \\ & & B_k & A_k^T \\ & & & B_{k+1} \end{bmatrix}, \tag{5}$$

the recurrence relations (3) and (4) may be rewritten as:

$$\begin{aligned} \overline{U}_{k+1}E_1B_1 &= B, \\ A\overline{V}_k &= \overline{U}_{k+1}T_k, \\ A^T\overline{U}_{k+1} &= \overline{V}_kT_k^T + V_{k+1}A_{k+1}E_{k+1}^T, \\ A^T\overline{U}_k &= \overline{V}_k\overline{T}_k^T, \end{aligned} \tag{6}$$

where $\overline{T}_k$ obtained from T_k by deleting its p last rows and E_i is the $(k+1)p \times p$ matrix which is zero except for the ith p rows, which are the $p \times p$ identity matrix. We have also $\overline{V}_k^T \overline{V}_k = I_{kp}$ and $\overline{U}_{k+1}^T \overline{U}_{k+1} = I_{(k+1)p}$, where I_l is the $l \times l$ identity matrix. More details about the block Golub–Kahan process can be found in [19].

3. The extended block Golub–Kahan algorithm

The algorithm proceeds by first running k steps of the block Golub–Kahan process with A^{-T}, and then continuing with m iterations of the block Golub–Kahan process with A, while maintaining orthogonalization among all generated vectors in the sequence. Given a matrix $C \in \mathbb{R}^{n\times p}$, by performing k steps of the Golub–Kahan procedure to the pair (A^{-T}, C), we have

$$U_1B_1 = C, \quad V_1A_1 = A^{-1}U_1, \tag{7}$$

$$\left.\begin{array}{l} U_{i+1}B_{i+1} = A^{-T}V_i - U_iA_i^T, \\ V_{i+1}A_{i+1} = A^{-1}U_{i+1} - V_iB_{i+1}^T, \end{array}\right\} \quad i = 1, 2, \ldots, k, \tag{8}$$

where $U_i, V_i \in \mathbb{R}^{n\times p}, B_i, A_i \in \mathbb{R}^{p\times p}$, and $U_1B_1, V_1A_1, U_{i+1}B_{i+1}, V_{i+1}A_{i+1}$ are the QR decomposition of the matrices $C, A^{-1}U_1, A^{-T}V_i - U_iA_i^T, A^{-1}U_{i+1} - V_iB_{i+1}^T$, respectively. By defining $\overline{U}_k, \overline{V}_k$ and T_k as in (5), the recurrence relations (7) and (8) may be rewritten as:

$$\begin{aligned} \overline{U}_{k+1}E_1B_1 &= C, \\ A^{-T}\overline{V}_k &= \overline{U}_{k+1}T_k, \\ A^{-1}\overline{U}_{k+1} &= \overline{V}_kT_k^T + V_{k+1}A_{k+1}E_{k+1}^T, \\ A^{-1}\overline{U}_k &= \overline{V}_k\overline{T}_k^T, \end{aligned} \tag{9}$$

where, as in (6), $\overline{T}_k$ obtained from T_k by deleting its p last rows and E_j is the $(k+1)p\times p$ matrix which is zero except for the jth p rows, which is the $p \times p$ identity matrix. We have also $\overline{U}_{k+1}^T\overline{U}_{k+1} = I_{(k+1)p}$ and $\overline{V}_k^T\overline{V}_k = I_{kp}$. We can easily show that $[U_1, U_2, \ldots, U_k]$ and $[V_1, V_2, \ldots, V_k]$ are the orthonormal basis of the subspaces $\mathcal{K}_k((AA^T)^{-1}, C)$ and $\mathcal{K}_k((A^TA)^{-1}, A^{-1}C)$, respectively. Now we again use the block Golub and Kahan bidiagonalization applied to the pair (A, U_1) in order to construct the matrices $Q_1, Q_2, \ldots, Q_m$ and $P_1, P_2, \ldots, P_{m+1}$ such that $\mathcal{U}_{k+1,m} = [U_1, U_2, \ldots, U_{k+1}, Q_1, Q_2, \ldots, Q_m]$ and $\mathcal{V}_{k,m+1} = [V_1, V_2, \ldots, V_k, P_1, P_2, \ldots, P_{m+1}]$ form the orthonormal basis of the subspaces $\mathcal{K}_{k+1,m}^e(AA^T, C)$ and $\mathcal{K}_{k,m+1}^e(A^TA, A^TC)$, respectively. In order to have the orthonormal basis $\mathcal{U}_{k+1,m}, \mathcal{V}_{k,m+1}$, first we orthogonal the matrix A^TU_1 against $V_1, V_2, \ldots, V_k$ and we generate the matrix P_1 satisfying:

$$P_1\tilde{A}_1 = A^TU_1 - \sum_{i=1}^{k} V_iH_{i1}, \tag{10}$$

where $P_1\tilde{A}_1$ is the QR decomposition of the matrix $A^TU_1 - \sum_{i=1}^{k} V_iH_{i1}$. Then, we orthogonal the matrix $AP_1 - U_1\tilde{A}_1^T$ against $U_2, U_3, \ldots, U_{k+1}$ and we generate the matrix Q_1 satisfying

$$Q_1\tilde{B}_1 = AP_1 - U_1\tilde{A}_1^T - \sum_{i=2}^{k+1} U_iG_{i1}, \tag{11}$$

where $Q_1\tilde{B}_1$ is the QR decomposition of the matrix $AP_1 - U_1\tilde{A}_1^T - \sum_{i=2}^{k+1} U_iG_{i1}$. Now we construct $Q_2, Q_3, \ldots, Q_{m+1}$ and $P_2, P_3, \ldots, P_{m+1}$ with the recurrence relations:

$$\left.\begin{array}{l} P_i\tilde{A}_i = A^TQ_{i-1} - P_{i-1}\tilde{B}_{i-1}^T, \\ Q_i\tilde{B}_i = AP_i - Q_{i-1}\tilde{A}_i^T, \end{array}\right\} \quad i = 2, 3, \ldots, m+1. \tag{12}$$

With the definitions $G_{11} = \tilde{A}_1^T$ and

$$\overline{Q}_m \equiv [Q_1, Q_2, \ldots, Q_m], \quad \overline{P}_m \equiv [P_1, P_2, \ldots, P_m], \quad \tilde{T}_m \equiv \begin{bmatrix} \tilde{B}_1^T & & & \\ \tilde{A}_2 & \tilde{B}_2^T & & \\ & \ddots & \ddots & \\ & & \tilde{A}_m & \tilde{B}_m^T \\ & & & \tilde{A}_{m+1} \end{bmatrix},$$

the recurrence relations (12) may be rewritten as

$$\begin{aligned} A^T\overline{Q}_m &= \overline{P}_{m+1}\tilde{T}_m, \\ A\overline{P}_m &= \overline{Q}_m\overline{\tilde{T}}_m^T + \left(\textstyle\sum_{i=1}^{k+1} U_iG_{i1}\right)E_1^T. \end{aligned} \tag{13}$$

where $\overline{\overline{T}}_m$ is the matrix obtained from $\tilde{T}_m$ by deleting p last rows and E_1 is the $mp \times p$ matrix which is zero except for the first p rows.

The main steps of the extended block Golub–Kahan algorithm to generate $\mathcal{U}_{k+1,m}$ and $\mathcal{V}_{k,m+1}$ may be summarized as follows.

Algorithm 1 The extended block Golub–Kahan algorithm

1. Inputs: $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times p}$, k, and m.
2. $U_1B_1 = C$, $V_1A_1 = A^{-1}U_1$ (QR decomposition of C and $A^{-1}U_1$),
3. For $i = 1, \ldots, k$
   - $W = A^{-T}V_i - U_iA_i^T$,
   - $U_{i+1}B_{i+1} = W$ (QR decomposition of W),
   - $W = A^{-1}U_{i+1} - V_iB_{i+1}^T$,
   - $V_{i+1}A_{i+1} = W$ (QR decomposition of W),

   end for.
4. $W = A^TU_1$,
   For $i = 1, \ldots, k$
   - $H_{i1} = V_i^TW$,
   - $W = W - V_iH_{i1}$,

   end for.
   $P_1\tilde{A}_1 = W$ (QR decomposition of W),
5. $W = AP_1 - U_1\tilde{A}_1^T$,
   For $i = 2, \ldots, k+1$
   - $G_{i1} = U_i^TW$,
   - $W = W - U_iG_{i1}$,

   end for.
   $Q_1\tilde{B}_1 = W$ (QR decomposition of W),
6. For $i = 2, \ldots, m+1$
   - $W = A^TQ_{i-1} - P_{i-1}\tilde{B}_{i-1}^T$,
   - $P_i\tilde{A}_i = W$ (QR decomposition of W),
   - $W = AP_i - Q_{i-1}\tilde{A}_i^T$,
   - $Q_i\tilde{B}_i = W$ (QR decomposition of W),

   end for.

The extended block Golub–Kahan algorithm will be breakdown if one of the matrices B_{i+1} (at step i of part 3), $\tilde{A}_1$ (in the computation of matrix P_1), and $\tilde{A}_i$ (at step i of part 6) of Algorithm 1 is singular. So the Algorithm 1 will not breakdown if all the matrices B_i, $i = 1, \ldots, k+1$ and $\tilde{A}_i$, $i = 1, \ldots, m+1$ are nonsingular. We will not treat the problem of breakdown in this paper and we assume that all the matrices B_i's and $\tilde{A}_i$'s produced by the extended block Golub–Kahan algorithm are nonsingular.

For the extended block Golub–Kahan Algorithm, we have the following propositions.

Proposition 1. *Suppose that (k, m) steps of the extended block Golub–Kahan Algorithm have been taken, then the matrices $\mathcal{U}_{k+1,m} = [U_1, U_2 \ldots, U_{k+1}, Q_1, Q_2, \ldots, Q_m]$ and $\mathcal{V}_{k,m+1} = [V_1, V_2 \ldots, V_k, P_1, P_2, \ldots, P_{m+1}]$ are the orthonormal bases of the extended block Krylov subspaces $\mathcal{K}^e_{k+1,m}(AA^T, C)$ and $\mathcal{K}^e_{k,m+1}(A^TA, A^TC)$, respectively.*

Proof. The proof of this proposition is similar to that given in [20] for the classical Arnoldi process. □

Proposition 2. *Suppose that (k,m) steps of Algorithm 1 have been carried out. Let*

$$F_{k+1} = \begin{bmatrix} I_p & A_1^TB_2^{-1} & & & \\ & I_p & A_2^TB_3^{-1} & & \\ & & I_p & \ddots & \\ & & & \ddots & A_k^TB_{k+1}^{-1} \\ & & & & I_p \end{bmatrix}, \quad J_k = \begin{bmatrix} H_{11} & B_2^{-1} & & & \\ H_{21} & & B_3^{-1} & & \\ \vdots & & & \ddots & \\ H_{k1} & & & & B_{k+1}^{-1} \end{bmatrix}. \tag{14}$$

Then we have

$$A^T\mathcal{U}_{k+1,m} = \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}, \qquad \text{with } \mathcal{F}_{k+1,m} = \begin{bmatrix} J_k F_{k+1}^{-1} & | & 0_{kp\times mp} \\ ------ & & ----- \\ \tilde{A}_1 E_1^T F_{k+1}^{-1} & | & \tilde{B}_1^T E_1^T \\ ------ & & ----- \\ 0_{mp\times(k+1)p} & | & \underline{\tilde{T}}_m \end{bmatrix}, \tag{15}$$

where $\underline{\tilde{T}}_m$ is the matrix obtained from $\tilde{T}_m$ by deleting its p first row and $E_1^T = [I_p, 0_p, \ldots, 0_p] \in \mathbb{R}^{p\times mp}$.

Proof. From (8) and (10), we have

$$A^T U_1 = P_1\tilde{A}_1 + \sum_{i=1}^{k} V_i H_{i1},$$
$$A^T U_{i+1} + A^T U_i A_i^T B_{i+1}^{-1} = V_i B_{i+1}^{-1}, \qquad \text{for } i = 1, \ldots, k.$$

By using the definition of matrices F_{k+1} and J_k, these equations can be written as follows:

$$A^T\overline{U}_{k+1}F_{k+1} = [\overline{V}_k, P_1]\begin{bmatrix} J_k \\ \tilde{A}_1 E_1^T \end{bmatrix},$$

where $E_1^T = [I_p, 0_p, \ldots, 0_p] \in \mathbb{R}^{p\times(k+1)p}$. This together with the first relation of (13) implies the desired relation (15). □

4. Low rank approximate solution to the continuous-time algebraic Riccati equation

Eq. (1) arises from the continuous-time linear-quadratic optimal control problem:

$$\text{Minimize } J(x_0, u) = \frac{1}{2}\int_0^{+\infty} (y(t)^T y(t) + u(t)^T u(t))dt, \tag{16}$$

subject to the dynamics constraints

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \quad x(0) = x_0, \end{cases} \tag{17}$$

where $x(t)$ is the state vector of dimension n, $u(t)$ is a control vector of $\mathbb{R}^s$, and $y(t)$ is the output of length p. Under the hypotheses [21]: the pair (A, B) is c-stabilizable (i.e., there is a matrix S such that $A - BS$ is stable) and the pair (C, A) is c-detectable (i.e., the pair (A^T, C^T) c-stabilizable), then $J(x_0, u)$ is minimized by $u(t) = -B^T Xx(t)$, where $X \in \mathbb{R}^{n\times n}$ is the unique symmetric positive semidefinite and stabilizing solution ($Re(\lambda(A - BB^T X) < 0)$) of the algebraic Riccati equation (1).

Many numerical methods have been proposed for the solution of (1), such as Newton type methods, eigenvector approaches; see, e.g., [8,9,11,22–24]. Most of the proposed methods are effective for relatively small problems. For large and sparse problems, projection methods onto block Krylov subspaces have been applied to compute low rank approximate solutions CAREs [13,14,25]. These methods usually require large projection subspaces and this increases considerably the CPU time and the memory requirements. To remedy the drawback of the projection methods based on the block or global Arnoldi algorithms, Heyouni and Jbilou in [12] introduced extended block Arnoldi process for solving approximate solution to (1).

The aim of this section is to show how to use the extended block Golub–Kahan algorithm described in Section 3 to extract low rank approximate solution to the continuous-time algebraic Riccati equation (1). This will be done by projecting the initial problem onto the extended block Krylov subspace $\mathcal{K}^e_{k+1,m}(AA^T, C^T)$. Applying the extended block Golub–Kahan Algorithm 1 to the pair (A, C^T) gives us the orthonormal basis $\mathcal{U}_{k+1,m} = [U_1, \ldots, U_{k+1}, Q_1, \ldots, Q_m]$ and $\mathcal{V}_{k,m+1} = [V_1, \ldots, V_k, P_1, \ldots, P_{m+1}]$ of the extended block Krylov subspaces $\mathcal{K}^e_{k+1,m}(AA^T, C^T)$ and $\mathcal{K}^e_{k,m+1}(A^TA, A^TC^T)$, respectively. In addition, by using Proposition 2, we can define the matrix

$$\mathcal{T}_{k+1,m} = \mathcal{U}_{k+1,m}^T A^T \mathcal{U}_{k+1,m} = \mathcal{U}_{k+1,m}^T \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}, \tag{18}$$

where the matrix $\mathcal{F}_{k+1,m}$ can be obtained through the algorithm. Using the orthonormal basis $\mathcal{U}_{k+1,m}$, as in [13,26], we look for low-rank approximate solution that have the form

$$X_{k+1,m} = \mathcal{U}_{k+1,m}Y_{k+1,m}\mathcal{U}_{k+1,m}^T, \tag{19}$$

where $Y_{k+1,m} \in \mathbb{R}^{(k+1+m)p\times(k+1+m)p}$. Using the expression (19) in Eq. (1), multiplying on the left by $\mathcal{U}_{k+1,m}^T$ and on the right by $\mathcal{U}_{k+1,m}$, we get the low-dimensional continuous-time algebraic Riccati equation

$$\mathcal{T}_{k+1,m}Y_{k+1,m} + Y_{k+1,m}\mathcal{T}_{k+1,m}^T - Y_{k+1,m}\bar{B}_{k+1,m}\bar{B}_{k+1,m}^T Y_{k+1,m} + \bar{C}_{k+1,m}^T\bar{C}_{k+1,m} = 0, \tag{20}$$

with $\bar{B}_{k+1,m} = \mathcal{U}_{k+1,m}^T B$ and $\bar{C}_{k+1,m}^T = \mathcal{U}_{k+1,m}^T C^T$. We assume that the low-dimensional continuous-time algebraic Riccati equation (20) has a unique symmetric positive semidefinite and stabilizing solution $Y_{k+1,m}$. The low-dimensional equation (20) can be solved by a standard direct method such as the Schur method [27].

Let $R_{k+1,m} = A^T X_{k+1,m} + X_{k+1,m}A - X_{k+1,m}BB^T X_{k+1,m} + C^T C$, be the residual associated with the approximate solution $X_{k+1,m}$. To stop the iterations, one has to test whether $\|R_{k+1,m}\|_F \le tol$, where tol is some fixed tolerance. The computation of $X_{k+1,m}$ (and of $R_{k+1,m}$) becomes expensive as the pair (k, m) increases. The next result shows that how to compute the residual without computing the approximation $X_{k+1,m}$ which is calculated only when convergence is achieved.

Theorem 3. *Let $C^T = U_1 B_1$ and $Y_{k+1,m}$ be the exact solution of (20) and $X_{k+1,m} = \mathcal{U}_{k+1,m} Y_{k+1,m} \mathcal{U}_{k+1,m}^T$ be the approximate solution to the continuous-time algebraic Riccati equation (1) obtained after (k, m) iterations of the extended block Golub–Kahan method. Then the residual $R_{k+1,m}$ associated to $X_{k+1,m}$ satisfies*

$$R_{k+1,m} = \overline{R}_{k+1,m} + \overline{R}_{k+1,m}^T, \tag{21}$$

where $\overline{R}_{k+1,m} = (\mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m} - \mathcal{U}_{k+1,m}\mathcal{T}_{k+1,m})Y_{k+1,m}\mathcal{U}_{k+1,m}^T$.

Proof. Starting from $R_{k+1,m} = A^T X_{k+1,m} + X_{k+1,m}A - X_{k+1,m}BB^T X_{k+1,m} + C^T C$ and using (15), we have

$$\begin{aligned} R_{k+1,m} &= \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}Y_{k+1,m}\mathcal{U}_{k+1,m}^T + \mathcal{U}_{k+1,m}Y_{k+1,m}\mathcal{F}_{k+1,m}^T\mathcal{V}_{k,m+1}^T \\ &\quad - \mathcal{U}_{k+1,m}Y_{k+1,m}\bar{B}_{k+1,m}\bar{B}_{k+1,m}^T Y_{k+1,m}\mathcal{U}_{k+1,m}^T + C^T C. \end{aligned}$$

From $\bar{C}_{k+1,m}^T = \mathcal{U}_{k+1,m}^T C^T = E_1 B_1$, where $E_1 \in \mathbb{R}^{(k+1+m)p\times p}$ is the matrix of the first p columns of the $(k+1+m)p \times (k+1+m)p$ identity matrix, we have

$$C^T C = U_1 B_1 B_1^T U_1^T = \mathcal{U}_{k+1,m} E_1 B_1 B_1^T E_1^T \mathcal{U}_{k+1,m}^T = \mathcal{U}_{k+1,m}\bar{C}_{k+1,m}^T \bar{C}_{k+1,m}\mathcal{U}_{k+1,m}^T$$

So, we get

$$\begin{aligned} R_{k+1,m} &= \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}Y_{k+1,m}\mathcal{U}_{k+1,m}^T + \mathcal{U}_{k+1,m}Y_{k+1,m}\mathcal{F}_{k+1,m}^T\mathcal{V}_{k,m+1}^T \\ &\quad + \mathcal{U}_{k+1,m}(-Y_{k+1,m}\bar{B}_{k+1,m}\bar{B}_{k+1,m}^T Y_{k+1,m} + \bar{C}_{k+1,m}^T\bar{C}_{k+1,m})\mathcal{U}_{k+1,m}^T. \end{aligned}$$

Since $Y_{k+1,m}$ is the symmetric solution of reduced CARE (20), this relation can be written as

$$\begin{aligned} R_{k+1,m} &= \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}Y_{k+1,m}\mathcal{U}_{k+1,m}^T + \mathcal{U}_{k+1,m}Y_{k+1,m}\mathcal{F}_{k+1,m}^T\mathcal{V}_{k,m+1}^T \\ &\quad - \mathcal{U}_{k+1,m}(\mathcal{T}_{k+1,m}Y_{k+1,m} + Y_{k+1,m}\mathcal{T}_{k+1,m}^T)\mathcal{U}_{k+1,m}^T, \end{aligned}$$

which yields the relation (21). □

We mention that the matrix $\mathcal{F}_{k+1,m}$ and $\mathcal{T}_{k+1,m}$ can be easily updated in each iteration. So, the norm of residual in each iteration can be computed cheaply without computing $X_{k+1,m}$.

By the experiments, we observe that, it is appropriate to take the size k for $\overline{U}_k$ (Golub–Kahan basis for the Krylov subspace $\mathcal{K}_k(AA^T, (AA^T)^{-k+1}C^T)$) small such as $k = 10$. For some fixed k, the extended block Golub–Kahan algorithm for the continuous-time algebraic Riccati equation (1) is summarized as follows.

Algorithm 2 The extended block Golub–Kahan algorithm for continuous-time algebraic Riccati equations

1. Inputs: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$, $C \in \mathbb{R}^{p\times n}$, the integers k, $mmax$, and a tolerance tol.
2. Generate the matrices $\overline{U}_{k+1} = [U_1, U_2, \ldots, U_{k+1}]$, $\overline{V}_{k+1} = [V_1, V_2, \ldots, V_{k+1}]$, T_k, Q_1, and P_1 using the steps 1–5 of Algorithm 1 with (A, C^T).
   Compute $J_k F_{k+1}^{-1}$, $\tilde{A}_1 E_1^T F_{k+1}^{-1}$ using Eq. (14).
3. For $m = 2, \ldots, mmax$
   - $W = A^T Q_{i-1} - P_{i-1}\tilde{B}_{i-1}^T$,
   - $P_i\tilde{A}_i = W$ (QR decomposition of W),
   - $W = AP_i - Q_{i-1}\tilde{A}_i^T$,
   - $Q_i\tilde{B}_i = W$ (QR decomposition of W),
   - Set $\mathcal{U}_{k+1,m} = [U_1, \cdots, U_{k+1}, Q_1, \cdots, Q_m]$ and $\mathcal{V}_{k,m+1} = [V_1, \cdots, V_k, P_1, \cdots, P_{m+1}]$.
   - Compute $\mathcal{F}_{k+1,m}$ using Eq. (15).
   - Compute $\bar{B}_{k+1,m} = \mathcal{U}_{k+1,m}^T B$, $\bar{C}_{k+1,m}^T = \mathcal{U}_{k+1,m}^T C^T$, and $\mathcal{T}_{k+1,m} = \mathcal{U}_{k+1,m}^T \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}$.
   - Solve by a direct method (the Schur method) the low-order Riccati equation
   $$\mathcal{T}_{k+1,m}Y_{k+1,m} + Y_{k+1,m}\mathcal{T}_{k+1,m}^T - Y_{k+1,m}\bar{B}_{k+1,m}\bar{B}_{k+1,m}^T Y_{k+1,m} + \bar{C}_{k+1,m}^T\bar{C}_{k+1,m} = 0.$$
   - Compute $R_{k+1,m}$ using Eq. (21). If $\|R_{k+1,m}\|_F < tol$, then compute the obtained approximation $X_{k+1,m} = \mathcal{U}_{k+1,m}Y_{k+1,m}\mathcal{U}_{k+1,m}^T$ and stop.

   End.

5. Low rank approximate solution to the differential matrix Riccati equations

We first recall the following theoretical result which gives under some assumptions on the coefficient matrices A, B, and C, an expression of the exact solution of (2), see [1] for more details.

Theorem 4. *Assuming that* (A, B) *is stabilizable and* (C, A) *is observable that* $X(0) > 0$, *the differential Riccati equation* (2) *admits a unique solution* X *given by*

$$X(t) = \widetilde{X} + e^{t\widetilde{A}^T}[e^{t\widetilde{A}^T}\widetilde{Z}e^{t\widetilde{A}^T} + (X_0 - \widetilde{X})^{-1} - \widetilde{Z}]^{-1}e^{t\widetilde{A}^T}, \tag{22}$$

where $\widetilde{X}$ *is the positive definite solution of algebraic Riccati equation*

$$A^T\widetilde{X} + \widetilde{X}A - \widetilde{X}BB^T\widetilde{X} + C^TC = 0, \quad \widetilde{A} = A - BB^T\widetilde{X}, \tag{23}$$

and $\widetilde{Z}$ *is the positive definite solution of the Lyapunov equation*

$$\widetilde{A}Z + Z\widetilde{A}^T - BB^T = 0.$$

The formula (22) is not suitable for large scale problems as it requires the computation of a matrix exponential, of an inverse matrix, and various products of matrices.

In this section, we show how to obtain low rank approximate solutions to the differential matrix Riccati equations (2) by projecting directly the initial problem onto small extended block Krylov subspace $\mathcal{K}^e_{k+1,m}(AA^T, C^T)$. As in Section 4, we look for low-rank approximate solution that have the form

$$X_{k+1,m}(t) = \mathcal{U}_{k+1,m}Y_{k+1,m}(t)\mathcal{U}^T_{k+1,m}, \qquad t \in [t_0, T_f], \tag{24}$$

and satisfying the Petrov–Galerkin orthogonality condition

$$\mathcal{U}^T_{k+1,m}R_{k+1,m}(t)\mathcal{U}_{k+1,m} = 0, \qquad t \in [t_0, T_f], \tag{25}$$

where $R_{k+1,m}(t)$ is the residual $R_{k+1,m}(t) = \dot{X}_{k+1,m}(t) - A^TX_{k+1,m}(t) - X_{k+1,m}(t)A + X_{k+1,m}(t)BB^TX_{k+1,m}(t) - C^TC$. From (24) and (25), we obtain the low dimensional differential matrix Riccati equation

$$\dot{Y}_{k+1,m}(t) - \mathcal{T}_{k+1,m}Y_{k+1,m}(t) - Y_{k+1,m}(t)\mathcal{T}^T_{k+1,m} + Y_{k+1,m}(t)\bar{B}_{k+1,m}\bar{B}^T_{k+1,m}Y_{k+1,m}(t) - \bar{C}^T_{k+1,m}\bar{C}_{k+1,m} = 0, \tag{26}$$

with $\bar{B}_{k+1,m} = \mathcal{U}^T_{k+1,m}B$ and $\bar{C}^T_{k+1,m} = \mathcal{U}^T_{k+1,m}C^T$. The latter low dimensional differential matrix Riccati equation is solved by using the well known Backward Differentiation Formula (BDF) method described in Section 5.1.

We assume that at each time t_l the approximate solution $X_{k+1,m}(t_l)$ can be given as a product of two low rank matrices. Consider the eigen-decomposition of the symmetric and positive definite matrix $Y_{k+1,m}(t_l) = \hat{U}D\hat{U}^T$, where D is the diagonal matrix of the eigenvalues of $Y_{k+1,m}(t_l)$ sorted in decreasing order. Let $\hat{U}_{m_l}$ be the $(m+k+1)p \times m_l$ matrix of the first m_l columns of $\hat{U}$ corresponding to the m_l eigenvalues of magnitude greater than some tolerance *dtol*. We obtain the truncated eigen-decomposition $Y_{k+1,m}(t_l) \approx \hat{U}_{m_l}D_{m_l}\hat{U}^T_{m_l}$, where $D_{m_l} = \mathrm{diag}[\lambda_1, \ldots, \lambda_{m_l}]$. Setting $\widetilde{Z}_{(k+1,m),l} = \hat{U}_{m_l}D^{1/2}_{m_l}$ and $Z_{(k+1,m),l} = \mathcal{U}_{k+1,m}\widetilde{Z}_{(k+1,m),l}$, it follows that

$$X_{k+1,m}(t_l) \approx Z_{(k+1,m),l}Z^T_{(k+1,m),l}. \tag{27}$$

The computation of $X_{k+1,m}(t)$ (and of $R_{k+1,m}(t)$) becomes expensive as k and m increase. So, in order to stop the iterations, one has to test if $\|R_{k+1,m}(t)\| < \epsilon$ without having to compute extra products involving the matrix A. The next result shows how to compute the residual norm of $R_{k+1,m}(t)$ without forming the approximation $X_{k+1,m}(t)$ which is computed in a factored form only when convergence is achieved.

Theorem 5. *Let* $C^T = U_1B_1$ *and* $Y_{k+1,m}(t)$ *be the exact solution of* (26) *and* $X_{k+1,m}(t) = \mathcal{U}_{k+1,m}Y_{k+1,m}(t)\mathcal{U}^T_{k+1,m}$ *be the approximate solution to the differential matrix Riccati equation* (2) *obtained after* (k, m) *iterations of the extended block Golub–Kahan method. Then the residual* $R_{k+1,m}(t)$ *associated to* $X_{k+1,m}(t)$ *satisfies*

$$R_{k+1,m}(t) = \overline{R}_{k+1,m}(t) + \overline{R}_{k+1,m}(t)^T, \tag{28}$$

where $\overline{R}_{k+1,m}(t) = (\mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m} - \mathcal{U}_{k+1,m}\mathcal{T}_{k+1,m})Y_{k+1,m}(t)\mathcal{U}^T_{k+1,m}$.

Proof. The proof is similar to that of Theorem 3. □

5.1. BDF for solving the low order differential matrix Riccati equation (26)

We use the Backward Differentiation Formula (BDF) method [28] for solving, at each step (k, m) of the extended block Golub–Kahan Algorithm 1, the low dimensional differential matrix Riccati equation (26). At each time t_l, let $Y_{(k+1,m),l}$ be

Table 1
Coefficients of p step BDF method with $p \leq 3$.

p	β	α_0	α_1	α_2
1	1	1		
2	2/3	4/3	−1/3	
3	6/11	18/11	−9/11	2/11

the approximation of $Y_{k+1,m}(t_l)$, where $Y_{k+1,m}$ is a solution of (26). Then, the new approximation $Y_{(k+1,m),l+1}$ of $Y_{k+1,m}(t_{l+1})$ obtained at step $l+1$ by BDF is defined by the implicit relation

$$Y_{(k+1,m),l+1} = \sum_{i=0}^{p-1} \alpha_i Y_{(k+1,m),l-i} + h\beta \mathfrak{F}(Y_{(k+1,m),l+1}), \tag{29}$$

where $h = t_{l+1} - t_l$ is the step size, α_i and β are the coefficients of the BDF method as listed in Table 1 and $\mathfrak{F}(Y)$ is given by

$$\mathfrak{F}(Y) = \mathcal{T}_{k+1,m} Y + Y \mathcal{T}_{k+1,m}^T - Y \bar{B}_{k+1,m} \bar{B}_{k+1,m}^T Y + \bar{C}_{k+1,m}^T \bar{C}_{k+1,m}.$$

The approximate $Y_{(k+1,m),l+1}$ solves the following matrix equation

$$\begin{aligned} &-Y_{(k+1,m),l+1} + h\beta(\mathcal{T}_{k+1,m} Y_{(k+1,m),l+1} + Y_{(k+1,m),l+1} \mathcal{T}_{k+1,m}^T \\ &\quad - Y_{(k+1,m),l+1} \bar{B}_{k+1,m} \bar{B}_{k+1,m}^T Y_{(k+1,m),l+1} + \bar{C}_{k+1,m}^T \bar{C}_{k+1,m}) + \sum_{i=0}^{p-1} \alpha_i Y_{(k+1,m),l-i} = 0. \end{aligned}$$

which can be written as the following continuous-time algebraic Riccati equation

$$\mathbb{T}_{k+1,m} Y_{(k+1,m),l+1} + Y_{(k+1,m),l+1} \mathbb{T}_{k+1,m}^T - Y_{(k+1,m),l+1} \mathbb{B}_{k+1,m} \mathbb{B}_{k+1,m}^T Y_{(k+1,m),l+1} + \mathbb{C}_{(k+1,m),l+1}^T \mathbb{C}_{(k+1,m),l+1} = 0. \tag{30}$$

By using the low rank product $Y_{(k+1,m),l} \approx \widetilde{Z}_{(k+1,m),l} \widetilde{Z}_{(k+1,m),l}^T$, $\widetilde{Z}_{(k+1,m),l} \in \mathbb{R}^{(k+m+1)p \times m_l}$, with $m_l < (k+m+1)p$ (which described in the previous section), the coefficient matrices appearing in (30) are given by

$$\begin{aligned} \mathbb{T}_{k+1,m} &= h\beta \mathcal{T}_{k+1,m} - \tfrac{1}{2} I, \\ \mathbb{B}_{k+1,m} &= \sqrt{h\beta} \bar{B}_{k+1,m}, \\ \mathbb{C}_{(k+1,m),l+1} &= [\sqrt{h\beta} \bar{C}_{k+1,m}, \sqrt{\alpha_0} Z_{(k+1,m),l}^T, \ldots, \sqrt{\alpha_{p-1}} \widetilde{Z}_{(k+1,m),l+1-p}^T]^T. \end{aligned}$$

The continuous-time algebraic Riccati equation (30) can be solved by applying direct methods based on Schur decomposition or based on generalized eigenvalues of the Hamiltonian in the small dimensional cases [9,22,24] or matrix sign function methods [29–31].

In order to initialize the BDF(p) integration scheme, the $p-1$ approximates $X_1, \ldots, X_{p-1}$ are computed by lower-order integration schemes. In our tests, we chose $p = 2$ and X_1 was computed as a product of low-rank factors ($X_1 \approx Z_1 Z_1^T$) by the Implicit Euler method BDF(1).

We summarize the steps of our proposed approach (using the extended block Golub–Kahan procedure) in Algorithm 3.

Remark. In the next section we compare the results obtained by the extended block Golub–Kahan and the block Golub–Kahan procedures. For the latter procedure, by defining

$$X_m = \overline{U}_m Y_m \overline{U}_m^T, \tag{31}$$

and using

$$\mathcal{T}_m = \overline{U}_m^T A^T \overline{U}_m = \overline{U}_m^T \overline{V}_m \overline{T}_m^T, \tag{32}$$

which obtained from the last relation of (6), we have the following low-dimensional equation

$$\mathcal{T}_m Y_m + Y_m \mathcal{T}_m^T - Y_m \bar{B}_m \bar{B}_m^T Y_m + \bar{C}_m^T \bar{C}_m = 0, \tag{33}$$

with $\bar{B}_m = \overline{U}_m^T B$ and $\bar{C}_m^T = \overline{U}_m^T C^T$. The residual R_m associated to X_m is as follows:

$$R_m = \overline{R}_m + \overline{R}_m^T, \tag{34}$$

where $\overline{R}_m = (\overline{V}_m \overline{T}_m^T - \overline{U}_m \mathcal{T}_m) Y_m \overline{U}_m^T$.

For the differential matrix Riccati equation, using the block Golub–Kahan procedure, the similar results can be given.

Algorithm 3 The extended block Golub–Kahan algorithm for differential continuous-time algebraic Riccati equations

1. Inputs: $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$, $C \in \mathbb{R}^{p\times n}$, the integers k, $mmax$, and a tolerance tol.
2. Generate the matrices $\overline{U}_{k+1} = [U_1, U_2, \ldots, U_{k+1}]$, $\overline{V}_{k+1} = [V_1, V_2, \ldots, V_{k+1}]$, T_k and vectors Q_1, P_1 using the steps 1–5 of Algorithm 1 with (A, C^T).
   Compute $J_k F_{k+1}^{-1}$, $\tilde{A}_1 E_1^T F_{k+1}^{-1}$ using Eq. (14).
   Compute $X_1, \ldots, X_{p-1}$ as low-rank products $X_j \approx Z_j Z_j^T$.
3. For $i = 2, \ldots, mmax$
   $W = A^T Q_{i-1} - P_{i-1}\tilde{B}_{i-1}^T$,
   $P_i\tilde{A}_i = W$ (QR decomposition of W),
   $W = AP_i - Q_{i-1}\tilde{A}_i^T$,
   $Q_i\tilde{B}_i = W$ (QR decomposition of W),
   Set $\mathcal{U}_{k+1,m} = [U_1, \cdots, U_{k+1}, Q_1, \cdots, Q_m]$ and $\mathcal{V}_{k,m+1} = [V_1, \cdots, V_k, P_1, \cdots, P_{m+1}]$.
   Compute $\mathcal{F}_{k+1,m}$ using Eq. (15).
   Compute $\bar{B}_{k+1,m} = \mathcal{U}_{k+1,m}^T B$, $\bar{C}_{k+1,m}^T = \mathcal{U}_{k+1,m}^T C^T$, and $\mathcal{T}_{k+1,m} = \mathcal{U}_{k+1,m}^T \mathcal{V}_{k,m+1}\mathcal{F}_{k+1,m}$.
   Use the BDF method to solve the low dimensional differential Riccati equation
   $$\dot{Y}_{k+1,m} - \mathcal{T}_{k+1,m}Y_{k+1,m} - Y_{k+1,m}\mathcal{T}_{k+1,m}^T + Y_{k+1,m}\bar{B}_{k+1,m}\bar{B}_{k+1,m}^T Y_{k+1,m} - \bar{C}_{k+1,m}^T\bar{C}_{k+1,m} = 0, \quad t \in [t_0, T_f].$$
   If $\|R_{k+1,m}\|_F < tol$, stop and compute the approximate solution $X_{k+1,m}(t)$ in the factored form given by the relation (27).
   End.

6. Numerical experiments

In this section, we report some experimental results. All the numerical experiments have been coded in MATLAB 2014a with windows 8 (64 bit) PC-Intel(R) Core(TM) i7-7700 CPU 3.60 GHz, 16 GB of RAM. The projected low-dimensional problem (20) was solved by using MATLAB functions care.m from MATLAB Toolbox. For Examples 1 and 2, we compare the performance of the extended block Golub–Kahan (EBGK-CARE) and the block Golub–Kahan (BGK-CARE) methods with equal-sized approximation spaces. In Tables 1 and 2, we give the number of iterations (Iter), the residual norm (Res. norm), and the CPU time in seconds (CPU time) required for convergence. For Example 3, we compare the performance of these methods by using the Krylov subspace $K_m(AA^T, C^T)$ and the extended Krylov subspace $\mathcal{K}_{k+1,m}^e(AA^T, C^T)$ with the same size subspace m.

Example 1. The matrix A is generated from the five-point discretization of the operator

$$L(u) = \Delta u - \sin(x+2y)\frac{\partial u}{\partial x} - e^y\frac{\partial u}{\partial y} - xy,$$

on the unit square $[0, 1] \times [0, 1]$ with homogeneous Dirichlet boundary conditions [14]. The number of inner grid points in each direction is n_0 and the dimension of the matrix A is $n = n_0^2$. For this experiment we set $n_0 = 50$, $s = 3$, and $p = 3$. The entries of the matrix B are random values uniformly distributed on the interval $[0, 1]$ and $C = I_{p\times n}$ is the identity $p \times n$ matrix. The results are shown in Table 2. The BGK-CARE (using $m = 50$) is compared to EBGK-CARE (using $m = 40, k + 1 = 10$ and $m = 44, k + 1 = 6$). Thus, the same size subspaces are used. These results indicate that the EBGK-CARE method is effective for this problem and it is better than BGK-CARE method with corresponding size subspace, in terms of the residual norm and the CPU time.

Example 2. This example is taken from [13]. The matrix A is of size $n = 1000$ and is given by:

$$A = -\begin{bmatrix} 4 & 0.5 & 0 & \ldots & 0 & 1 \\ 1.5 & 4 & 0.5 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & 0.5 \\ 1 & 0 & \ldots & 0 & 1.5 & 4 \end{bmatrix}.$$

The entries of the matrix B have random values uniformly distributed on $[0, 1]$ and the number of the columns in B is $s = 2$ and the matrix $C = I_{p\times n}$ is the identity $p \times n$ matrix, where $p = 2$. Using the BGK-CARE, the EBGK-CARE, and the EBA-CARE (Extended Block Arnoldi [12]) methods with the same size subspaces $m1 = 16$, we obtained the results presented in Table 3. The results in Table 3 illustrate that the EBGK-CARE method clearly outperforms the BGK-CARE

Table 2
Numerical results for Example 1.

Method	Iter.	Res. norm	CPU time
BGK-CARE	$m = 50$	$8.7646e-04$	5.73
EBGK-CARE	$m = 40,\ k+1 = 10$	$2.5193e-05$	4.31
	$m = 44,\ k+1 = 6$	$8.0725e-05$	4.71

Table 3
Numerical results for Example 2.

Method	Iter.	Res. norm	CPU time
BGK-CARE	$m = 16$	$2.1604e-08$	0.32
EBGK-CARE	$m = 10,\ k+1 = 6$ $(m1 = m+k+1 = 16)$	$1.4115e-12$	0.30
EBA-CARE	$m = 8$ $(m1 = 2m = 16)$	$5.9065e-12$	0.15

Table 4
Numerical results for Example 3.

Size (A)	Method	Iter.	Res. norm	CPU time
49×49	BGK-BDF(2)	$m = 14$	$1.9297e-06$	0.05
	EBGK-BDF(2)	$m = 14,\ k+1 = 7$	$1.3086e-10$	0.15
81×81	BGK-BDF(2)	$m = 15$	$4.9565e-06$	0.04
	EBGK-BDF(2)	$m = 15,\ k+1 = 10$	$4.0781e-09$	0.17
100×100	BGK-BDF(2)	$m = 15$	$1.1833e-05$	0.03
	EBGK-BDF(2)	$m = 15,\ k+1 = 10$	$5.1146e-08$	0.17

method in terms of residual norm and computation time. The EBA-CARE method needs less CPU time than EBGK-CARE method and they reach the same accuracy in terms of the residual norm.

Example 3. This example was taken from [32] and comes from the autonomous linear-quadratic optimal control problem of one dimensional heat flow

$$\begin{aligned}
&\frac{\partial}{\partial t}x(t,\gamma) = \frac{\partial^2}{\partial \gamma^2}x(t,\gamma) + b(\gamma)u(t),\\
&x(t,0) = x(t,1) = 0,\ t > 0,\\
&x(0,\gamma) = x_0,\ \gamma \in [0,1],\\
&y(x) = \int_0^1 c(\gamma)x(t,\gamma)d\gamma,\ x > 0.
\end{aligned}$$

Using a standard finite element approach based on the first order B-splines, we obtain the following ordinary differential equation

$$\begin{cases} M\dot{X}(t) = KX(t) + FU(t),\\ y(t) = CX(t), \end{cases}$$

where the matrices M and K are given by:

$$M = \frac{1}{6n}\begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ & & & 1 & 4 \end{bmatrix}, \quad K = -\alpha n \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

Using the semi-implicit Euler method, we get the following discrete dynamical system

$$(M - \Delta tK)\dot{x} = Mx(t) + \Delta tFU_k.$$

We set $A = -(M - \Delta tK)^{-1}M$ and $B = \Delta t(M - \Delta tK)^{-1}F$. The entries of the $n \times p$ matrix F and the $p \times n$ matrix C are given random values uniformly distributed on [0, 1]. We chose the initial condition as $X_0 = 0_{n\times n} = Z_0Z_0^T$, where $Z_0 = 0_{n\times 2}$ and we set $p = 2$, $\alpha = 0.05$, and $\Delta t = 0.01$. In Table 4, we reported the number of iterations, residual norms, and the run times for various sizes of A. In this table, the parameter m presents the dimension of the Krylov subspace $K_m(AA^T, C^T)$ which can be used in the BGK-BDF(2) method for obtaining the most accurate approximate solution for this example. Table 4 shows that, by using the EBGK-BDF(2) method, we can obtain more accurate approximate solution than the one obtained by the BGK-BDF(2) method at little extra cost.

7. Conclusion

In the present paper, we have described the extended version of block Golub–Kahan procedure and its properties. By using the extended block Golub–Kahan procedure, we have presented a new projection method for computing low rank approximate solutions for large-scale algebraic and differential matrix Riccati equations. We gave some theoretical results for the residual at each step which does not require the computation of products of large matrices. Finally, some numerical experiments were given in order to compare the block Golub–Kahan and extended block Golub–Kahan procedures.

Acknowledgments

The authors are very much indebted to the referees for their constructive comments and suggestions which greatly improved the original manuscript of this paper.

References

[1] B.D.O. Anderson, J.B. Moore, Linear Optimal Control, Prentice-Hall, Englewood Cliffs, NJ, 1971.
[2] P. Benner, J. Li, T. Penzl, Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems, Numer. Linear Algebra Appl. 15 (2008) 755–777.
[3] D.A. Bini, B. Iannazzo, B. Meini, Numerical Solution of Algebraic Riccati Equations, SIAM, Philadelphia, 2012.
[4] S. Bittanti, A. Laub, J.C. Willems, The Riccat Equation, Springer-Verlag, Berlin, 1991.
[5] M.J. Corless, A.D. Frazho, Linear systems and control, an operator perspective, in: Monographs and Textbooks in Pure and Applied Mathematics, vol. 254, Marcel Dekker, New York and Basel, 2003.
[6] B.N. Datta, Numerical Methods for Linear Control Systems Design and Analysis, Elsevier Academic Press, Amsterdam, 2003.
[7] Zoran Gajić, M.T.J. Qureshi, Lyapunov matrix equation in system stabiliy and control, in: Mathematics in Science and Engineering, vol. 195, Associated Press, San Diego and London, 1995.
[8] P. Lancaster, L. Rodman, Algebraic Riccati Equations, Oxford University Press, Oxford, 1995.
[9] V. Mehrmann, The autonomous linear quadratic control problem: Theory and numerical solution, in: Lecture Notes in Control and Information Sciences, vol. 163, Springer-Verlag, Heidelberg, 1991.
[10] K. Zhou, J.C. Doyle, K. Glover, Robust and Optimal Control, Prentice-Hall, Upper saddle River, 1995.
[11] P. Benner, R. Byers, An exact line search method for solving generalized continuous algebraic Riccati equations, IEEE Trans. Automat. Control 43 (1) (1998) 101–107.
[12] M. Heyouni, K. Jbilou, An extended block Arnoldi algorithm for large-scale solutions of the continous-time algebraic Riccati equation, Electron. Trans. Numer. Anal. 33 (2009) 53–62.
[13] K. Jbilou, Block Krylov subspace methods for large continous-time algebraic Riccati equations, Numer. Algorithms 34 (2003) 339–353.
[14] K. Jbilou, An Arnoldi based algorithm for large algebraic Riccati equations, Appl. Math. Lett. 19 (2006) 437–444.
[15] V. Simoncini, A new iterative method for solving large-scale Lyapunov matrix equations, SIAM J. Sci. Comput. 29 (3) (2007) 1268–1288.
[16] P. BENNER, H. MENA, BDF methods for large-scale differential Riccati equations, in: Proceedings of Mathematical Theory of Network and Systems, MTNS 2004, 2004.
[17] V. Hernández, J.J. Ibánez, J. Peinado, E. Arias, A GMRES-based BDF method for solving differential Riccati equations, Appl. Math. Comput. 196 (2008) 613–626.
[18] G.H. Golub, W. Kahan, Algorithm LSQR is based on the Lanczos process and bidiagonalization procedure, SIAM J. Numer. Anal. 2 (1965) 205–224.
[19] S. Karimi, F. Toutounian, The block least squres method for solving nonsymmetric linear systems with multiple rigth- hand sides, Appl. Math. Comput. 177 (2006) 852–862.
[20] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.
[21] W.M. Wonham, On a matrix Riccati equation of stochastic control, SIAM J. Control 6 (1968) 681–697.
[22] W.F. Arnold, III, A.J. Laub, Generalized eigenproblem algorithms and software for algebraic Riccati equations, Proc. IEEE 72 (1984) 1746–1754.
[23] J.J. Hench, A.J. Laub, Numerical solution of the discrete-time periodic Riccati equation, IEEE Trans. Automat. Control 39 (1994) 1197–1209.
[24] P. Van Dooren, A generalized eigenvalue approach for solving Riccati equations, SIAM J. Sci. Stat. Comput. 2 (1981) 121–135.
[25] I.M. Jaimoukha, E.M. Kasenally, Krylov subspace methods for solving large Lyapunov equations, SIAM J. Numer. Anal. 31 (1994) 227–251.
[26] Y. Saad, Numerical solution of large Lyapunov equations, in: M.A. Kaashoak, J.H. van Schuppen, A.C.M. Ran (Eds.), Signal Processing, Scattering, Operator Theory and Numerical Methods, Birkhauser, Basel, 1990, pp. 503–511.
[27] A.J. Laub, A Schur method for solving algebraic Riccati equations, IEEE Trans. Automat. Control 24 (1979) 913–921.
[28] C. Choi, A.J. Laub, Constructing Riccati differential equations with known analytic solutions for numerical experiments, IEEE Trans. Automat. Control 35 (1990) 437–439.
[29] R. Byers, Solving the algebraic Riccati equation with the matrix sign function, Linear Algebra Appl. 85 (1987) 267–279.
[30] C.S. Kenny, A.J. Laub, P. Papadopoulos, Matrix sign function algorithms for Riccati equations, in: Proceedings of IMA Conference on Control: Modelling, Computation, Information, IEEE Computer Society Press, Southend-On-Sea, 1992, pp. 1–10.
[31] D.J. Roberts, Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, Internt. J. Control. 32 (1998) 677–687.
[32] Y. Güğan, M. Hached, K. Jbilou, M. Kurulay, Low rank approximate solutions to large-scale differential matrix Riccati equations. systems, 2017, arXiv:161200499v2 (Accessed on 11 Apr 2017).